

**SISTEM INFORMASI PENJADWALAN PRODUKSI *COLT*
T120SS MENGGUNAKAN METODE *MASTER PRODUCTION*
SCHEDULE UNTUK OPTIMASI PENJADWALAN PRODUKSI
PADA PT KRAMA YUDHA RATU MOTOR
TUGAS AKHIR**

Diajukan Untuk Memenuhi Sebagian Syarat Penyelesaian Program Sarjana
Terapan Program Studi Sistem Informasi Industri Otomotif
Pada Politeknik STMI Jakarta

OLEH
FRANS WILLIAM JEREMIA HASIAN
1314008



POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Frans William Jeremia Hasian
 NIM : 1314008
 Judul Tugas Akhir : Sistem Informasi Penjadwalan Produksi *Colt* T120ss
 Menggunakan Metode *Master Production Schedule*
 (MPS) untuk Optimasi Penjadwalan Produksi Pada PT
 Krama Yudha Ratu Motor
 Pembimbing : Lucky Heriyanto, S.T, M.T.I.

Tanggal	Keterangan	Paraf
18 Mei 2018	Bimbingan BAB I	
28 Mei 2018	Revisi BAB I	
6 Juni 2018	Bimbingan BAB II	
7 Juni 2018	Revisi BAB II	
2 Juli 2018	Bimbingan BAB III	
4 Juli 2018	Revisi BAB III	
6 Juli 2018	Revisi BAB III	
10 Juli 2018	Bimbingan BAB IV	
12 Juli 2018	Revisi BAB IV	
19 Juli 2018	Bimbingan BAB V	
3 Agustus 2018	Revisi BAB V	
6 Agustus 2018	Revisi BAB V	
7 Agustus 2018	Revisi BAB V & BAB VI	
7 Agustus 2018	Revisi BAB V	
7 Agustus 2018	ACC Seminar	

Mengetahui,
 Ketua Program Studi
 Sistem Informasi Industri Otomotif

Dosen Pembimbing

Drs. Jacob Saragih, M.M.
 NIP : 195404281986031002

Lucky Heriyanto, S.T, M.T.I.
 NIP : 197908202009011009

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Frans William Jeremia Hasian

NIM : 1314019

Berstatus sebagai mahasiswa Program Studi Sistem Informasi Industri Otomotif di Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

“SISTEM INFORMASI PENJADWALAN PRODUKSI COLT T120SS MENGGUNAKAN METODE MASTER PRODUCTION SCHEDULE UNTUK OPTIMASI PENJADWALAN PRODUKSI PADA PT KRAMA YUDHA RATU MOTOR”

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, *survey* lapangan, dibantu oleh dosen pembimbing maupun asisten dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, 8 Agustus 2018

Frans William Jeremia Hasian

ABSTRAK

Penjadwalan produksi merupakan hal utama dalam proses bisnis suatu perusahaan khususnya pada industri manufaktur. Meskipun diakui bahwa kualitas produk mempunyai pengaruh yang sangat besar terhadap kelancaran penjualan, akan tetapi penjadwalan produksi yang menentukan jalannya suatu proses produksi. Dari hal tersebut, diperoleh permasalahan seperti belum adanya sistem yang dapat mengatur proses penjadwalan produksi yang berada di PT Krama Yudha Ratu Motor. Proses perhitungan penjadwalan produksi yang masih dilakukan dengan menggunakan kalkulator menyebabkan kesulitan dalam menghitung penjadwalan produksi yang dibutuhkan. Untuk memecahkan masalah tersebut, dibutuhkan sistem yang dapat menghitung jadwal produksi secara otomatis. Sistem informasi ini dibuat menggunakan metode pengembangan sistem *waterfall*. Metode ini diawali dari identifikasi kebutuhan, analisis kebutuhan, merancang sistem dengan UML, merancang data dengan ERD, dan melakukan perancangan antar muka. Sistem ini dibangun menggunakan MySQL 5.0.11 dan PHP 5.6.30. Salah satu metode yang dibutuhkan untuk mendapatkan penjadwalan produksi yang baik adalah menggunakan *Master Production Schedule*. MPS merupakan perencanaan produksi jangka pendek pada suatu perusahaan yang berisi tentang rencana menyeluruh serta perinciannya dalam menghasilkan produk akhir (produk jadi). Dengan adanya sistem yang dibangun ini, maka diperoleh sistem informasi penjadwalan produksi yang dapat membantu proses penjadwalan, dan dapat melakukan perhitungan otomatis penjadwalan produksi.

Kata Kunci : Sistem Informasi, *Master Production Schedule*, *Waterfall*,
Penjadwalan Produksi

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	ii
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING.....	iii
LEMBAR PERTANYAAN KEASLIAN	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Pokok Permasalahan.....	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah.....	3
1.5. Manfaat Tugas Akhir.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. Konsep Dasar Sistem.....	6
2.1.1. Karakteristik Sistem.....	7
2.1.2. Klasifikasi Sistem	10
2.1.3. Jenis Sistem	11
2.2. Konsep Dasar <i>Master Production Schedule</i> (MPS).....	12
2.2.1. Pengertian <i>Master Production Schedule</i>	12
2.2.2. Perhitungan <i>Master Production Schedule</i>	13
2.3. Konsep Penjadwalan Produksi.....	14

2.3.1.	Pengertian Penjadwalan Produksi.....	14
2.3.2.	Tujuan Penjadwalan Produksi	14
2.3.3.	Manfaat Penjadwalan Produksi	15
2.4.	<i>System Development Life Cycle (SDLC)</i>	16
2.5.	Model <i>Waterfall</i>	20
2.6.	Analisis PIECES	22
2.7.	<i>System Requirements</i>	23
2.8.	<i>Flowmap</i>	24
2.9.	<i>Unified Model Language (UML)</i>	26
2.9.1.	<i>Use Case Diagram</i>	27
2.9.2.	<i>Use Case Description</i>	28
2.9.3.	<i>Activity Diagram</i>	29
2.9.4.	<i>Sequence Diagram</i>	31
2.9.5.	<i>Class Diagram</i>	32
2.10.	<i>Database</i>	33
2.11.	<i>Entity Relationship Diagram (ERD)</i>	33
2.12.	Kamus Data.....	34
2.13.	<i>Windows Navigation Diagram</i>	35
2.14.	<i>Hypertext Preprocessor (PHP)</i>	36
2.15.	XAMPP	37
2.16.	MySQL	37
2.16.1.	Tipe Data MySQL	38
2.17.	<i>Blackbox Testing</i>	39
BAB III	METODOLOGI PENELITIAN	40
3.1.	Metodologi Penelitian	40
3.2.	Pengumpulan Data	40
3.3.	Metode Pengumpulan Data.....	41
3.4.	Metode Pengembangan Sistem	41
3.5.	<i>Literature Review</i>	43
3.6.	Kerangka Pemecah Masalah	46

BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	50
4.1.	Tentang Perusahaan.....	50
4.2.	Profil Perusahaan	52
4.3.	Visi dan Misi Perusahaan	53
4.4.	Manajemen Mutu	54
4.5.	Struktur Organisasi Perusahaan	54
4.6.	Struktur Organisasi Divisi PPIC	56
4.7.	Analisis Dokumen Sistem Penjadwalan Produksi	59
	4.7.1. Dokumen Masuk	59
	4.7.2. Dokumen Keluar	61
4.8.	Analisis Proses (<i>Flowmap</i> Sistem yang Sedang Berjalan)	63
4.9.	<i>Flowmap</i> Proses Penjadwalan Produksi	63
4.10.	Analisis Permasalahan.....	65
BAB V	ANALISIS DAN PEMBAHASAN	67
5.1.	Alur Proses Penjadwalan Produksi Usulan.....	69
5.2.	Analisis Kebutuhan Sistem.....	67
	5.2.1. <i>Functional Requirement</i>	70
	5.2.2. <i>Non Functional Requirement</i>	71
5.3.	<i>Use Case Diagram</i>	71
	5.3.1. <i>Use Case Description</i>	72
5.4.	<i>Activity Diagram</i>	77
5.5.	<i>Sequence Diagram</i>	84
5.6.	<i>Class Diagram</i>	89
5.7.	<i>Deployment Diagram</i>	91
5.8.	<i>Windows Navigation Diagram</i> (WND).....	91
5.9.	Pemodelan Data Sistem Usulan	93
	5.9.1. <i>Entity Relationship Diagram</i> (ERD).....	93
	5.9.2. Kamus Data.....	94
5.10.	Perancangan <i>Interface</i> Sistem Informasi Penjadwalan Produksi.....	96

BAB VI	PENUTUP	106
6.1.	Kesimpulan	106
6.2.	Saran	106
Daftar Pustaka	107
Lampiran		

DAFTAR GAMBAR

		Halaman
Gambar II.1	Gambaran Sistem Terbuka	11
Gambar II.2	Gambaran Sistem Tertutup.....	12
Gambar II.3	Metode <i>Waterfall</i>	20
Gambar II.4	Contoh <i>Windows Navigation Diagram</i>	33
Gambar III.1	Kerangka Penelitian	49
Gambar IV.1	PT Krama Yudha Ratu Motor.....	52
Gambar IV.2	Struktur Organisasi Induk.....	55
Gambar IV.3	Struktur Organisasi Divisi PPIC PT Krama Yudha Ratu Motor	56
Gambar IV.4	<i>Form CKD Simulation</i>	59
Gambar IV.5	<i>Form Man Hour</i>	60
Gambar IV.6	<i>Form Master Production Schedule</i>	61
Gambar IV.7	<i>Form Daily Production Schedule</i>	62
Gambar IV.8	<i>Flowmap</i> Proses Penjadwalan Induk Produksi.....	64
Gambar V.1	<i>Flowmap</i> Proses Penjadwalan Produksi Usulan.....	68
Gambar V.2	<i>Use Case</i> Usulan Sistem Informasi Penjadwalan Produksi	72
Gambar V.3	<i>Activity Diagram Login</i>	77
Gambar V.4	<i>Activity Diagram</i> Mengelola CKD <i>Simulation</i>	78
Gambar V.5	<i>Activity Diagram</i> Mengelola Data Penjadwalan Produksi	79
Gambar V.6	<i>Activity Diagram</i> Melakukan Validasi <i>Forecast Order</i>	80
Gambar V.7	<i>Activity Diagram</i> Mengelola Data <i>Master Barang</i>	81
Gambar V.8	<i>Activity Diagram</i> Mencetak Laporan Produksi	82
Gambar V.9	<i>Activity Diagram</i> Mengelola Hak Akses.....	83
Gambar V.10	<i>Sequence Diagram Login</i>	84
Gambar V.11	<i>Sequence Diagram</i> Mengelola CKD <i>Simulation</i>	85
Gambar V.12	<i>Sequence Diagram</i> Mengelola Data Penjadwalan Produksi	86
Gambar V.13	<i>Sequence Diagram</i> Melakukan Validasi <i>Forecast Order</i>	87
Gambar V.14	<i>Sequence Diagram</i> Melakukan Validasi <i>Forecast Order</i>	87

Gambar V.15 <i>Sequence Diagram</i> Mengelola Data <i>Master</i> Barang.....	87	Gambar V.16 <i>Sequence Diagram</i> Mengelola Data <i>Master</i> Barang.....	87
Gambar V.17 <i>Sequence Diagram</i> Mencetak Laporan Produksi.....	88		
Gambar V.18 <i>Sequence Diagram</i> Mengelola Hak Akses	89		
Gambar V.19 <i>Class Diagram</i> Sistem Informasi Penjadwalan Produksi.....	90		
Gambar V.20 <i>Deployment Diagram</i> Sistem Informasi Penjadwalan Produksi	91		
Gambar V.21 <i>Windows Navigation Diagram</i> Sistem Informasi Penjadwalan Produksi.....	92		
Gambar V.22 <i>Entity Relationship Diagram</i> Sistem Informasi Penjadwalan Produksi.....	93		
Gambar V.23 Rancangan <i>Form Login</i>	96		
Gambar V.24 Rancangan Menu Utama	97		
Gambar V.25 Rancangan Data Barang	98		
Gambar V.26 Rancangan <i>Form</i> Tambah Barang	98		
Gambar V.27 Rancangan <i>Form</i> Ubah Barang.....	99		
Gambar V.28 Rancangan Data <i>Login</i>	99		
Gambar V.29 Rancangan <i>Form</i> Tambah Data <i>Login</i>	100		
Gambar V.30 Rancangan <i>Form</i> Ubah Data <i>Login</i>	100		
Gambar V.31 Rancangan Data CKD	101		
Gambar V.32 Rancangan <i>Form</i> Tambah CKD	101		
Gambar V.33 Rancangan <i>Form</i> Ubah CKD.....	102		
Gambar V.34 Rancangan Data Produksi.....	102		
Gambar V.35 Rancangan <i>Form</i> Tambah Data Produksi.....	103		
Gambar V.36 Rancangan <i>Form</i> Ubah Data Produksi.....	103		
Gambar V.37 Rancangan <i>Form</i> Rekap Produksi.....	104		
Gambar V.38 Rancangan Informasi Rekap Produksi	104		
Gambar V.39 Rancangan Informasi Cetak Produksi.....	105		

DAFTAR TABEL

		Halaman
Tabel II.1	Simbol-simbol <i>Flowmap</i>	25
Tabel II.2	Jenis Diagram Pada UML dan Fungsi-fungsinya	26
Tabel II.3	Simbol-simbol <i>Use Case Diagram</i>	28
Tabel II.4	Simbol-simbol <i>Activity Diagram</i>	30
Tabel II.5	Simbol-simbol <i>Sequence Diagram</i>	31
Tabel II.6	Simbol-simbol <i>Class Diagram</i>	32
Tabel II.7	Contoh Kamus Data untuk Tabel Pemasok	35
Tabel II.8	Simbol-simbol ERD	36
Tabel II.9	Tipe Data MySQL.....	38
Tabel III.1	<i>Literature Review</i>	44
Tabel IV.1	Analisis Permasalahan.....	65
Tabel V.1	Analisis K	65
Tabel V.2	<i>Use Case Description Login</i>	73
Tabel V.3	<i>Use Case Description Mengelola CKD Simulation</i>	73
Tabel V.4	<i>Use Case Description Mengelola Data Penjadwalan Produksi</i>	74
Tabel V.5	<i>Use Case Description Melakukan Validasi Forecast Order</i>	74
Tabel V.6	<i>Use Case Description Mengelola Data Master Barang</i>	75
Tabel V.7	<i>Use Case Description Merekap Laporan Produksi Bulanan</i>	75
Tabel V.8	<i>Use Case Description Mencetak Laporan Produksi Bulanan</i> ..	76
Tabel V.9	<i>Use Case Description Mengelola Hak Akses</i>	77
Tabel V.10	Tabel Data <i>Login</i>	94
Tabel V.11	Tabel Data <i>Barang</i>	94
Tabel V.12	Tabel Data <i>CKD Simulation</i>	95
Tabel V.13	Tabel Data <i>Produksi</i>	95
Tabel V.14	Tabel Data <i>Perubahan Forecast Order</i>	96

DAFTAR TABEL

	Halaman
Lampiran A Jadwal Wawancara.....	L-1
Lampiran B Jadwal Observasi.....	L-6
Lampiran C Contoh Perhitungan MPS	L-11
Lampiran D <i>Coding</i>	L-12
Lampiran E <i>Blackbox Testing</i>	L-36
Lampiran F <i>Screenshot</i> Program	L-40

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penjadwalan produksi merupakan hal utama dalam proses bisnis suatu perusahaan khususnya pada industri manufaktur. Meskipun diakui bahwa kualitas produk mempunyai pengaruh yang sangat besar terhadap kelancaran penjualan, akan tetapi penjadwalan produksi yang menentukan jalannya suatu proses produksi. Hal ini disebabkan jika dicermati kembali ternyata penjadwalan mempunyai banyak fungsi di antaranya menentukan berapa banyak jumlah produk yang harus dihasilkan untuk memenuhi permintaan pelanggan.

Era globalisasi yang semakin kompleks ini menyebabkan terjadinya persaingan yang semakin maju dengan perkembangan teknologi canggih. Oleh karena itu, setiap perusahaan harus memiliki sistem yang dapat berjalan sesuai dengan perkembangan tersebut. Salah satu proses yang membutuhkan sistem yaitu proses penjadwalan produksi pada suatu perusahaan. Perkembangan sistem dapat membantu suatu perusahaan dalam melakukan penjadwalan produksi pada perusahaan tersebut sehingga pelaksanaan produksi tidak terhambat dikarenakan ada masalah yang ditimbulkan karena tidak adanya sistem yang digunakan untuk melakukan proses penjadwalan.

Salah satu metode yang digunakan dalam penjadwalan produksi yaitu *master production schedule*. *Master Production Schedule* adalah perencanaan produksi jangka pendek pada suatu perusahaan yang berisi tentang rencana menyeluruh serta perinciannya dalam menghasilkan produk akhir (produk jadi). Dalam *Master Production Schedule* juga memuat prioritas model produk yang akan diproduksi, jadwal pembelian bahan-bahan produksi, jadwal pelaksanaan proses produksi dan jadwal kerja karyawan serta jadwal operasional mesin. *Master Production Schedule* ini juga bermanfaat dalam merencanakan kapasitas produksi dan kebutuhan material untuk aktivitas produksi. Hal ini yang membuat metode MPS banyak digunakan pada suatu perusahaan sebagai metode penjadwalannya.

PT Krama Yudha Ratu Motor (KRM) merupakan sebuah perseroan terbatas yang bergerak dalam bidang perakitan kendaraan bermotor jenis niaga salah satunya Mitsubishi *Colt* T120ss kendaraan jenis truk yang dimana produk paling populer buatan PT Krama Yudha Ratu Motor. Departemen PPIC (*Production Planning Inventory Control*) adalah salah satu departemen dalam struktur organisasi PT Krama Yudha Ratu Motor. Departemen ini memiliki peranan yang sangat penting dalam manajemen persediaan barang setengah jadi dan juga bertanggung jawab untuk perencanaan produksi. Departemen ini juga mempunyai *Sub Department Production Planning* yang menangani proses pembuatan jadwal induk produksi.

Prosedur penjadwalan produksi pada PT Krama Yudha Ratu Motor sudah terkomputerisasi, namun dalam memproses data, dan penyimpanan data masih menggunakan data fisik. Data yang berhubungan dengan proses penjadwalan produksi masih belum terintegrasi, sehingga terkadang menimbulkan masalah dalam pencarian data, *penginputan*, dan pengolahan data. Kesulitan dalam mengolah data penjadwalan produksi menyebabkan hasil penghitungan penjadwalan produksi yang tidak sesuai dengan metode yang ada.

Berdasarkan latar belakang dan permasalahan tersebut, maka disusun tugas akhir yang membahas tentang penjadwalan produksi. tugas akhir ini memiliki judul “SISTEM INFORMASI PENJADWALAN PRODUKSI COLT T120SS MENGGUNAKAN METODE *MASTER PRODUCTION SCHEDULE* UNTUK OPTIMASI PENJADWALAN PRODUKSI PADA PT KRAMA YUDHA RATU MOTOR”. Tugas akhir ini berfungsi untuk membantu perusahaan dalam proses penjadwalan produksi sehingga proses tersebut dapat dilakukan secara maksimal.

1.2 Pokok Permasalahan

Berdasarkan latar belakang yang dipaparkan sebelumnya, maka dapat diidentifikasi permasalahan-permasalahan yang terjadi pada proses penjadwalan produksi di PT Krama Yudha Ratu Motor adalah sebagai berikut:

1. Belum ada sistem yang dapat mengatur proses penjadwalan produksi yang berada di PT Krama Yudha Ratu Motor sehingga dapat terjadi kehilangan data, dan inkonsistensi data penjadwalan produksi.

2. Proses perhitungan penjadwalan produksi yang masih dilakukan dengan menggunakan kalkulator menyebabkan kesulitan dalam menghitung penjadwalan produksi yang dibutuhkan, hal ini terjadi disebabkan tidak semua *staff* pada bagian PPIC mengerti cara menghitung proses penjadwalan produksi.

1.3 Tujuan Penelitian

Tujuan penelitian dari tugas akhir ini yang dilakukan terhadap departemen PPIC PT Krama Yudha Ratu Motor dalam melakukan penjadwalan produksi adalah untuk merancang dan membangun sistem informasi penjadwalan produksi yang mampu:

1. Membangun sistem informasi penjadwalan produksi yang menyediakan fasilitas penyimpanan data dokumen dan data produksi dengan menggunakan *database* agar dapat disimpan di satu lokasi sehingga dapat membantu dalam proses pencarian data jika dibutuhkan.
2. Membangun sistem informasi penjadwalan produksi yang dapat melakukan perhitungan penjadwalan produksi.

1.4 Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan lebih terarah, maka perlu diadakan batasan-batasan sebagai berikut:

1. Tempat melakukan penelitian adalah di PT Krama Yudha Ratu Motor.
2. Penelitian dilakukan pada departemen PPIC selama satu bulan terhitung sejak 1 Agustus 2017 s.d. 31 Agustus 2017.
3. Ruang lingkup yang diamati sebatas pada departemen PPIC dan 1 produk yaitu *Colt T120ss* dalam proses penjadwalan produksi.
4. Sistem ini hanya dirancang bangun, akan tetapi belum diimplementasikan di dalam perusahaan.
5. Sistem ini dibangun menggunakan aplikasi XAMPP dengan versi PHP 5.6.30 dan *database* MySQL 5.0.11

1.5 Manfaat Tugas Akhir

Adapun manfaat bagi perusahaan yang dapat diambil dari penelitian Tugas Akhir ini untuk Perusahaan yaitu memiliki aplikasi penjadwalan produksi yang sudah berbasis *web*. Hal ini dapat membantu dalam melakukan penjadwalan produksi.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya. Adapun sistematika penulisan Tugas Akhir ini terurai dalam enam bab, yaitu:

BAB I PENDAHULUAN

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat tugas akhir, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini membahas tentang berbagai teori yang diperoleh dari buku-buku literatur ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Teori-teori yang dipaparkan pada laporan ini adalah seputar sistem informasi, *Unified Modelling Language* (UML), dan teori-teori lain yang berhubungan dengan perancangan sistem informasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metode ilmiah dalam mencari, mengembangkan, dan menguji kebenaran tentang suatu pengetahuan. Selain itu dijelaskan pula kerangka pemecahan masalah yang menguraikan tahap-tahap untuk mengatasi permasalahan yang telah diidentifikasi.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini akan membahas tentang data yang telah diperoleh berdasarkan penelitian di PT Krama Yudha Ratu Motor, meliputi profil perusahaan, struktur organisasi, proses bisnis sistem

penjadwalan produksi yang berjalan, dan dokumen yang terlibat pada proses bisnis tersebut.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi perancangan, meliputi perancangan sistem menggunakan *Unified Model Language*(UML), perancangan data menggunakan *Entity Relationship Diagram*(ERD), dan perancangan antar muka menggunakan *Windows Navigation Diagram*(WND)

BAB VI PENUTUP

Bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan, serta mengemukakan saran-saran dalam penerapan sistem informasi penjadwalan produksi untuk perusahaan dalam penelitian dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Dalam pengertian umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka. Dengan kata lain sistem juga merupakan sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Menurut Sutanto (dalam Djahir dan Pratita, 2014) Sistem adalah kumpulan atau grup dari subsistem atau bagian atau komponen apapun baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu. Berikut beberapa pengertian sistem dari beberapa sumber:

1. Menurut Mcleod (dalam Yakub, 2012:1), Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan.
2. Menurut Gordon B. Davis (dalam Rusdiana, 2014:28), Sistem merupakan bagian-bagian yang beroperasi secara bersama-sama untuk mencapai beberapa tujuan.
3. Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sutabri (2012:16).

Lebih lanjut menurut Puspitawati dan Anggadini (2011:2), Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut:

Sistem adalah kumpulan dari elemen-elemen untuk mencapai suatu tujuan tertentu. Sesuatu dapat dikatakan sistem apabila memenuhi dua syarat:

1. Memiliki bagian-bagian yang saling berintegrasi dengan maksud untuk mencapai suatu tujuan, bagian-bagian itu dinamakan sub-sistem.
2. Harus memenuhi tiga unsur *Input-Process-Output*.

2.1.1 Karakteristik Sistem

Karakteristik sistem menurut Sutanto yang dikutip dari buku sistem informasi akuntansi oleh Puspitawati dan Anggadini (2011:2), adalah adanya tujuan sistem, batas sistem, subsistem, hubungan sistem, lingkungan sistem, dan *input*, proses, dan *output*. Untuk lebih jelasnya akan diuraikan sebagai berikut:

1. Tujuan Sistem

Tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar target tersebut dapat tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian. Jadi kalau melihat kepada alasan diatas maka kriteria suatu tujuan itu mutlak adanya, tanpa ada kriteria yang jelas maka akan sulit dilakukan pengendalian karena sulit dilakukan pengukuran sehingga pada akhirnya sistem tidak akan mencapai sasaran. Untuk mengatasi masalah ini, bagi sebagian pelaku sistem, beberapa hal telah dilakukan yang didasari oleh pemikiran bahwa mungkin masa depan atau tujuan dari suatu sistem yang sedang dibangun atau dijalankan merupakan pengalaman masa lalu dari *user* atau pelaku sistem yang lain. Oleh karena itu di dalam menentukan tujuan dari suatu sistem harus diupayakan melibatkan orang-orang yang memiliki pengalaman yang sesuai dengan tujuan sistem yang akan dibangun untuk menghindari adanya penyimpangan baik dalam perancangan maupun pelaksanaannya.

2. Batas Sistem

Batas sistem merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya. Batas sistem ini bagi umat manusia sangat relatif dan tergantung pada tingkat pengetahuan dan situasi kondisi yang dirasakan oleh orang yang melihat sistem tersebut. Batas sistem yang mampu dibayangkan oleh seseorang akan sangat berbeda dengan sistem sebenarnya dalam dunia nyata. Karena itu batas sistem ini akan memberikan konsekuensi yang kurang baik seandainya dipaksakan untuk sama bagi setiap orang sebab

selain akan menghambat kreativitas pelaku sistem juga akan memperlambat evolusi dari sistem tersebut.

3. Subsistem

Subsistem merupakan komponen atau bagian dari suatu sistem, bisa fisik atau abstrak. Suatu subsistem akan memiliki subsistem yang lebih kecil dan seterusnya. Istilah yang menggambarkan bagian dari suatu sistem tidak harus selalu subsistem, istilah lain mungkin adalah komponen, elemen atau unsur. Bagi orang kedokteran mungkin lebih suka menggunakan istilah organ, bagi orang akuntansi mungkin lebih biasa menggunakan istilah prosedur, subsistem istilah yang digunakan untuk menunjukan bagian dari sistem pada saat sistem tersebut menjadi konteks pembahasan. Pada saat konteks pembahasan berpindah ke subsistem maka substansi tersebut menjadi sistem dan didalam sistem ini juga terdapat subsistem yang lebih kecil, sedang sistem yang sebelumnya dimana subsistem yang kini menjadi sistem berada, disebut sebagai supersistem. Dengan beralihnya konteks yang menjadi pembahasan atau dengan beralihnya bidang yang dianggap sebagai sistem maka beralih pula batas-batasnya.

4. Hubungan dan Hierarki Sistem

Hubungan sistem adalah hubungan yang terjadi antara subsistem dengan subsistem lainnya yang setingkat atau antara subsistem dengan sistem yang lebih besar. Ada dua macam hubungan sistem, yaitu hubungan horizontal dan hubungan vertikal. Hubungan sistem dengan subsistem yang terlihat digambar disebut dengan hubungan hierarki yang menggambarkan tingkatan. Sedangkan hubungan horizontal menunjukkan hubungan antara subsistem dengan subsistem lain yang setingkat.

5. *Input-Proses-Output*

Tiga komponen sistem fungsi/subsistem adalah *Input-Proses-Output*, fungsi ini juga menunjukkan bahwa sistem sebagai proses tidak bisa berdiri sendiri harus terdapat *input* dan *output*. Tiga komponen tersebut dapat dijabarkan sebagai berikut:

- a. *Input* merupakan segala bentuk masukan ke dalam suatu sistem, *input* bervariasi bisa berupa energi, manusia, data, model, dan lain-lain. *Input* merupakan pemicu bagi sistem untuk melakukan proses. *Input* dapat diklasifikasikan ke dalam tiga kategori sebagai berikut:
- 1) *Serial input* merupakan *input* yang diperoleh sebagai hasil atau *output* sistem sebelumnya.
 - 2) *Probable input* merupakan potensi *input* yang dapat digunakan oleh suatu sistem. Suatu sistem harus dapat menentukan *input* mana yang sesuai untuk menghasilkan *output* yang diharapkan.
 - 3) *Feedback* dan pengendalian, *feedback* adalah data atau informasi tentang kinerja (*output* dari suatu sistem). Terdapat beberapa jenis *feedback* seperti, *feedback input* yaitu merupakan *input* jenis ke tiga, *input* ini merupakan bagian *output* dari sistem yang sama yang digunakan sebagai *control*. *Feedback input* ini dikelompokkan lagi menjadi dua yaitu:
 - a) *Negative feedback input* digunakan sebagai alat kontrol untuk memperkecil. *Negative feedback* ini terus berlangsung sampai terjadi titik keseimbangan dimana antara *output* yang diharapkan dan realisasi terjadi kesesuaian dengan tingkat penyimpangan yang dapat diterima.
 - b) *Positive feedback input* digunakan sebagai alat kontrol untuk memperbesar. *Positive feedback* ini terus berlangsung sampai terjadi titik keseimbangan dimana antara *output* yang diharapkan dan realisasi terjadi kesesuaian dengan tingkat penyimpangan yang dapat diterima.
- b. Fungsi kedua yaitu Proses yang merupakan perubahan dari *input* menjadi *output*. Proses mungkin berupa perakitan yang menghasilkan satu macam *output* dari berbagai macam *input* yang disusun berdasarkan aturan tertentu.

c. Fungsi ketiga yaitu *Output*. *Output* adalah hasil dari suatu proses yang merupakan tujuan dari keberadaan sistem. *Output* dapat diklasifikasikan menjadi tiga macam yaitu:

- 1) *Output* yang langsung diberikan ke konsumen untuk dikonsumsi.
- 2) *Output* suatu sistem yang dikonsumsi oleh suatu subsistem yang lain dalam sistem yang sama dalam suatu siklus produksi.
- 3) *Output* yang merupakan bagian dari *output* secara keseluruhan yang dapat dikonsumsi oleh sistem yang lain atau oleh sistem yang bersangkutan, namun menjadi tidak berguna apabila dibuang ke lingkungan.

6. Lingkungan sistem

Lingkungan sistem adalah faktor-faktor diluar sistem yang mempengaruhi sistem. Lingkungan sistem ada dua macam yaitu lingkungan eksternal (lingkungan yang berada diluar sistem) dan lingkungan internal (lingkungan yang berada didalam suatu sistem).

2.1.2 Klasifikasi Sistem

Menurut Rusdiana dan Irfan (2014) sistem dapat diklasifikasikan dari beberapa sudut pandang yaitu sebagai berikut:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologi, yaitu suatu sistem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, sistem produksi, sistem penggajian, sistem penjualan, sistem administrasi personalia, dan lain sebagainya.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan

human machine system. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

4. Sistem Terbuka dan Sistem Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.1.3 Jenis Sistem

Jenis sistem secara umum terdiri atas sistem terbuka dan sistem tertutup. Sistem terbuka adalah sistem yang tidak memiliki sasaran, pengendalian mekanis dan umpan balik. Adapun sistem yang tertutup adalah sebuah sistem yang memiliki sasaran, pengendalian mekanis dan umpan balik (Rusdiana dan Irfan, 2014).

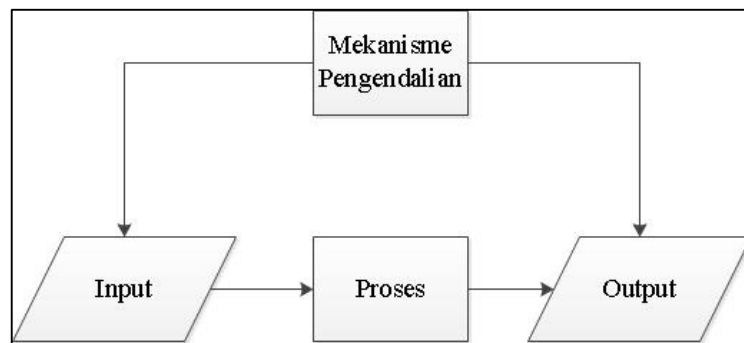
Secara sederhana, sistem terbuka dapat digambarkan sebagai berikut:



Gambar II.1 Gambaran Sistem Terbuka

(Sumber: Rusdiana dan Irfan, 2014)

Sistem tertutup adalah sistem yang tidak berinteraksi secara langsung dengan lingkungannya melalui arus sumber daya. Skema sistem tertutup dapat dilihat pada gambar di bawah ini.



Gambar II.2 Gambaran Sistem Tertutup

(Sumber: Rusdiana dan Irfan, 2014)

2.2 Konsep Dasar *Master Production Schedule*(MPS)

Master Production Schedule atau yang biasa juga disebut Jadwal Induk Produksi merupakan salah satu fungsi manajemen. Dalam MPS dapat di tentukan langkah-langkah yang perlu diambil oleh pimpinan untuk mencapai tujuan perusahaan dengan mempertimbangkan masalah-masalah yang mungkin timbul dimasa yang akan datang. Hasil dari MPS adalah sebuah rencana kerja dimana merupakan alternatif yang baik untuk mencapai tujuan yang ditetapkan. MPS merupakan pernyataan akhir mengenai berapa banyak item item akhir yang harus diproduksi dan kapan harus diproduksi.

2.2.1 Pengertian *Master Production Schedule*

Master Production Schedule (MPS) merupakan suatu pernyataan tentang produk akhir dari suatu perusahaan yang merencanakan memproduksi output berkaitan dengan kuantitas dan periode waktu. Aktifitas penjadwalan produksi induk pada dasarnya berkaitan dengan bagaimana menyusun dan memperbaharui jadwal produksi induk, memproses transaksi dari MPS, memelihara catatan-catatan MPS, mengevaluasi efektifitas dari MPS dan memberikan laporan evaluasi dalam periode waktu yang teratur untuk keperluan umpan balik dan tinjauan ulang (Gasperz, 2001).

Master Production Schedule dapat dikatakan sebagai modul sentral di dalam perencanaan dan kontrol sistem manufaktur. Sebuah *Master Production Schedule* yang efektif menghasilkan sebuah basis untuk memanfaatkan sumber daya manufaktur sebaik mungkin, memastikan pesanan pembeli terpenuhi, menyelesaikan masalah penjualan antara sales dan kegiatan produksi, dan mencapai tujuan strategis perusahaan sebagaimana yang ditampikan di dalam perencanaan operasi dan penjualan. *Master Production Schedule* adalah pernyataan tentang keluaran pada masa mendatang yang direncanakan dengan membuat spesifikasi produk-produk (atau produk yang dipilih) yang akan dibuat. Sebuah MPS menghasilkan spesifikasi tentang bagaimana produk akan disimpan di persediaan untuk memenuhi permintaan pada periode mendatang.

2.2.2 Perhitungan *Master Production Schedule*

Di dalam sebuah *Master Production Schedule*, ada beberapa perhitungan (Hartono, 2014) antara lain :

1. *Forecast*

Forecast merupakan perhitungan untuk meramalkan permintaan pada periode mendatang. Menggunakan rumus *Moving Average Forecasting* pada pesanan 6 minggu sebelumnya. Rumusnya adalah sebagai berikut:

$$6 - \text{period MAF} = \sum_{i=t-6+1}^t \text{Actual demand}_i / n \quad \dots\dots\dots(2.1)$$

2. *Master Production Schedule* (MPS)

MPS adalah jumlah komponen yang diproduksi pada satu periode. Nilai MPS berbeda tergantung dari kebijakan perusahaan. Pada PT Krama Yudha Ratu Motor dapat dilihat pada jumlah pesanan setiap bulannya.

3. *Projected Available Balance* (PAB)

Projected Available Balance (PAB) digunakan untuk menghitung perkiraan kondisi persediaan setelah dikurangi oleh peramalan permintaan (*forecast*).

4. *Available to Promise (ATP)*

ATP adalah perhitungan jumlah persediaan yang bisa dijanjikan untuk permintaan dihitung dari jumlah produksi (MPS) dikurangi jumlah terbesar dari permintaan (*demand*) atau peramalan (*forecast*). Rumusnya adalah sebagai berikut:

a. Untuk periode pertama

$$ATP = On\ Hand + MPS - Sum\ Of\ the\ orders\ until\ the\ next \\ \dots\dots\dots(2.2)$$

b. Untuk setiap periode MPS berikutnya

$$ATP = MPS - Sum\ of\ the\ orders\ until\ the\ next \dots\dots\dots(2.3)$$

2.3 Konsep Penjadwalan Produksi

Konsep penjadwalan produksi meliputi pengertian penjadwalan produksi, tujuan penjadwalan produksi, manfaat penjadwalan produksi, teknik-teknik penjadwalan produksi.

2.3.1 Pengertian Penjadwalan Produksi

Berikut menjelaskan mengenai definisi *scheduling* (penjadwalan) produksi menurut beberapa ahli:

1. *Scheduling* adalah penjadwalan kegiatan. Suatu kegiatan dijadwalkan kapan memulainya, berapa lama mengerjakan setiap tahap kegiatannya dan akhirnya kapan selesainya (Subagyo, 2005).
2. Penjadwalan (*scheduling*) produksi merupakan koordinasi waktu dalam kegiatan produksi untuk mengalokasikan bahan baku, bahan pembantu dan perlengkapan produksi pada fasilitas atau bagian pengolahan dalam pabrik sesuai dengan waktu yang telah ditetapkan, sehingga proses produksi dapat berjalan dengan lancar (Assauri, 2008).
3. Penjadwalan produksi adalah aktifitas pengalokasian sumber daya perusahaan untuk memproduksi suatu barang atau jasa dengan biaya dan tingkat persediaan yang rendah (Heizer dan Render, 2008).
4. Penjadwalan produksi adalah pengalokasian sumber daya yang terbatas

untuk mengerjakan sejumlah pekerjaan (Ginting, 2009).

2.3.2 Tujuan Penjadwalan Produksi

Mengidentifikasi beberapa tujuan dari aktivitas *scheduling* sebagai berikut (Ginting, 2009):

1. Meningkatkan penggunaan sumber daya atau mengurangi waktu tunggu, sehingga total waktu proses dapat berkurang dan produktivitasnya dapat meningkat.
2. Mengurangi persediaan barang setengah jadi atau mengurangi sejumlah pekerjaan yang menunggu dalam antrean ketika sumber daya yang ada masih mengerjakan tugas yang lain. Teori Baker mengatakan jika aliran kerja suatu jadwal konstan, maka aliran antrean yang mengurangi rata-rata waktu alir akan mengurangi rata-rata persediaan barang setengah jadi.
3. Mengurangi beberapa keterlambatan pada pekerjaan yang mempunyai batas waktu penyelesaian sehingga akan meminimalisir biaya keterlambatan.
4. Membantu pengambilan keputusan mengenai perencanaan kapasitas pabrik dan jenis kapasitas yang dibutuhkan sehingga penambahan biaya dapat dihindarkan.

2.3.3 Manfaat Penjadwalan Produksi

Menurut Heizer dan Render (2008) akan pentingnya penjadwalan adalah sebagai berikut:

1. Dengan *scheduling* yang efektif, perusahaan menggunakan asetnya dengan efektif dan menghasilkan kapasitas modal yang diinvestasikan menjadi lebih besar, yang sebaliknya akan mengurangi biaya.
2. *Scheduling* menambah kapasitas dan fleksibilitas yang terkait memberikan waktu pengiriman yang lebih cepat dan dengan demikian pelayanan kepada pelanggan menjadi baik.
3. Keuntungan yang ketiga dari bagusnya penjadwalan adalah keunggulan kompetitif dengan pengiriman yang bisa diandalkan.

Fungsi pokok dari *scheduling* adalah untuk membuat arus barang dapat

berjalan lancar sesuai dengan waktu yang direncanakan melalui produksi. Hal ini membuat penyelesaian dan pengiriman produk akan dilakukan tepat waktu, dan secara tidak langsung menghindari keterlambatan penerimaan kepada konsumen.

2.4 *System Development Life Cycle (SDLC)*

Menurut Dennis et al (2015) *System Development Life Cycle (SDLC)* memiliki 4 perangkat fase dasar yaitu *planning, analysis, design, dan implementation*. Beberapa proyek dalam membangun sebuah sistem informasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing, tetapi hampir semua proyek memiliki elemen dari 4 fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang menghasilkan *deliverable* / hasil kegiatan contohnya beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek.

Dalam beberapa proyek, tahapan dan langkah-langkah SDLC diproses dalam suatu alur tahapan, dimulai dari awal hingga akhir. Dalam proyek lain, tim proyek terus berpindah dari fase ke fase tersebut secara teratur, bertahap, iteratif, atau dalam bentuk pola lainnya. Dalam SDLC, dijelaskan fase-fase, tindakan, serta beberapa teknik yang digunakan untuk menyempurnakan langkah-langkah yang dijelaskan secara umum. Ada dua hal penting untuk dipahami tentang SDLC:

1. Mendapatkan dan memahami pengertian umum dari tahapan dan langkah-langkah yang digunakan dalam dalam proyek Sistem Informasi serta beberapa teknik yang menghasilkan dokumen yang pasti.
2. Kedua hal penting untuk dipahami bahwa SDLC merupakan proses penyempurnaan yang dilakukan secara bertahap.

Dokumen yang dihasilkan dalam tahap analisis, memberikan ide umum dari suatu bagian dari sistem baru. Dokumen dari *deliverable* ini digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan untuk menghasilkan dokumen yang menjelaskan secara detail dari sistem yang dibuat. Dokumen ini, akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Berikut dibawah ini merupakan penjabaran dari setiap fase dalam SDLC:

1. *Planning* (Perencanaan)

Tahap *planning* / perencanaan adalah proses dasar yang menjelaskan mengapa sebuah sistem informasi harus dibangun dan menentukan bagaimana sebuah tim proyek akan membangunnya. Terdapat dua langkah dalam *planning* (perencanaan) yaitu:

- a. Inisiasi proyek, pada tahap ini nilai bisnis suatu sistem terhadap organisasi diidentifikasi: Apakah sistem tersebut akan menurunkan biaya atau meningkatkan pendapatan? Sebagian besar gagasan untuk sistem baru berasal dari luar area Departemen IT (misalnya dari Departemen Pemasaran, Departemen Akuntansi) dalam suatu bentuk *system request* (permintaan sistem). Permintaan sistem menyajikan ringkasan singkat tentang kebutuhan sistem dalam suatu bisnis dan menjelaskan bagaimana suatu sistem akan menciptakan nilai bisnis. Departemen IT bekerja sama dengan departemen lain yang menghasilkan permintaan (disebut sponsor proyek) untuk melakukan analisis kelayakan sistem informasi. *System request* (permintaan sistem) dan analisis kelayakan disajikan dan dipresentasikan kepada pihak komite persetujuan *Information System* (komite pengarah), yang menentukan apakah proyek tersebut harus diambil atau tidak.
- b. Setelah proyek disetujui, proyek memasuki manajemen proyek. Selama manajemen proyek, manajer proyek membuat sebuah rencana kerja, membentuk anggota staff proyek, dan anggota teknisi lainnya di tempat, untuk membantu tim proyek mengendalikan dan mengarahkan proyek pada keseluruhan tahap SDLC.

2. *Analysis* (Analisis)

Analisis menurut Muslihudin dan Oktafianto (2016) merupakan tahapan awal dalam pengembangan sistem dan merupakan tahap fundamental yang sangat menentukan kualitas sistem yang akan dikembangkan. Tahap analisis dalam SDLC menjelaskan siapa yang akan memakai sistem, apa yang sistem kerjakan serta kapan dan dimana sistem akan digunakan. Selama pada tahap ini, tim proyek menyelidiki sistem yang ada saat ini,

mengidentifikasi peluang untuk melakukan perbaikan, dan mengembangkan sistem baru. Tahap ini memiliki tiga langkah:

- a. Langkah yang pertama adalah Strategi Analisis. Langkah ini dikembangkan untuk mengarahkan usaha dari tim proyek untuk menganalisis sistem yang telah ada (disebut sebagai *as-is system*) beserta masalah-masalah yang ada dan untuk merancang suatu sistem yang baru (disebut sebagai *to-be system*).
- b. Langkah selanjutnya adalah mengumpulkan kebutuhan-kebutuhan sistem (melalui wawancara atau kuesioner). Pada tahap ini terdapat juga masukan analisis informasi dari sponsor proyek dan pihak lainnya yang mengarah pada pengembangan konsep untuk sistem baru. Konsep sistem ini kemudian digunakan sebagai dasar untuk mengembangkan seperangkat model analisis bisnis, yang menggambarkan bagaimana bisnis akan beroperasi jika sistem baru dikembangkan.
- c. Kemudian Analisis, konsep sistem, dan model digabungkan menjadi dokumen yang disebut proposal sistem, yang dipresentasikan ke sponsor proyek dan pengambil keputusan utama lainnya (misalnya, anggota komite persetujuan) yang memutuskan apakah proyek tetap terus dilanjutkan.

3. *Design* (Perancangan)

Tahap *design* atau perancangan memutuskan bagaimana sistem akan beroperasi, mulai dari hal yang berkaitan dengan perangkat keras (*hardware*), perangkat lunak (*software*) dan infrastruktur jaringan seperti:

- a. Antarmuka pengguna (*interface*),
- b. Formulir,
- c. Laporan,
- d. Program spesifik: *database*, dan *file-file* lainnya yang akan dibutuhkan.

Meskipun sebagian besar keputusan strategis mengenai sistem dibuat dalam pengembangan konsep sistem selama tahap analisis, langkah-langkah dalam tahap desain menentukan secara tepat bagaimana sistem akan beroperasi. Fase desain ini memiliki empat langkah:

- a. Strategi perancangan (*Design Strategy*) adalah hal yang pertama kali yang dikembangkan. Strategi perancangan menjelaskan apakah sistem akan dikembangkan oleh programmer perusahaan sendiri, atau akan di-*outsourcing* ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket perangkat lunak yang sudah ada.
- b. Pengembangan desain arsitektur (*architecture design*) yang mengarah pada dasar untuk sistem, yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan. Dalam banyak kasus, sistem akan menambah atau mengubah infrastruktur yang sudah ada dalam suatu organisasi. Desain antarmuka menentukan bagaimana pengguna akan berpindah ke sistem yang baru (misalnya, metode navigasi seperti menu dan tombol di layar) serta laporan yang akan digunakan oleh sistem.
- c. Pengembangan *database* dan spesifikasi *file*. Pengembangan ini menentukan dengan tepat data apa yang akan disimpan dan di mana data tersebut akan disimpan.
- d. Tim analis mengembangkan desain program (*Program Design*), yang mendefinisikan program yang perlu dikembangkan dan apa yang akan dilakukan masing-masing program.

Kumpulan dari hasil kegiatan ini seperti desain arsitektur, desain antarmuka, spesifikasi *database* dan *file*, dan perancangan program merupakan spesifikasi sistem yang diserahkan ke tim pemrograman untuk diimplementasi. Pada akhir tahap desain, analisis kelayakan dan rencana proyek diperiksa ulang dan diperbaiki, dan keputusan lain dibuat oleh sponsor proyek dan komite persetujuan mengenai apakah proyek dihentikan atau dilanjutkan.

4. *Implementation* (Implementasi)

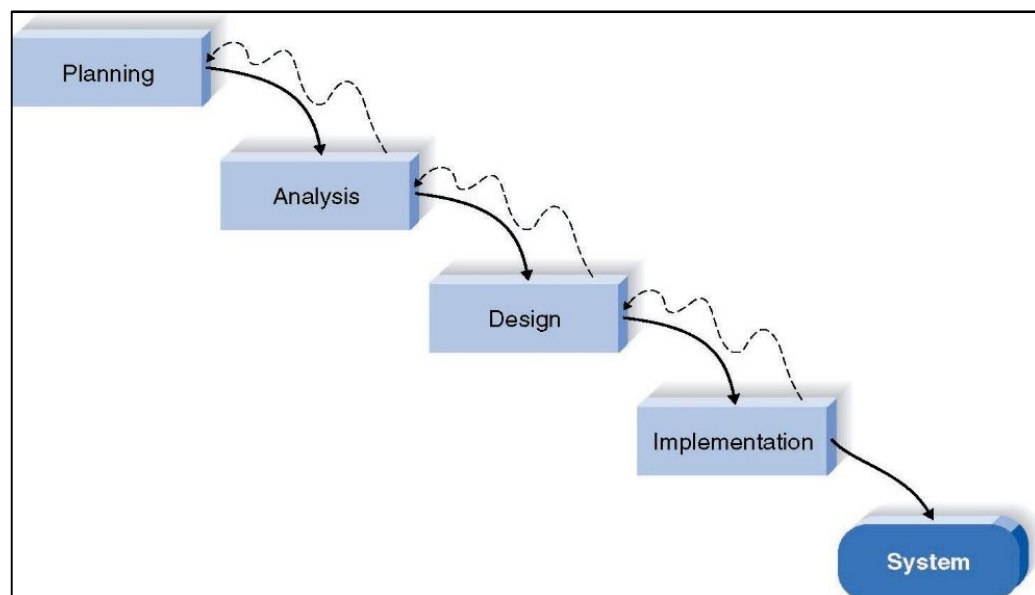
Tahap akhir dalam SDLC adalah tahap implementasi, di mana sistem benar-benar dibangun (atau dibeli, dalam kasus paket *desaign software*). Fase ini sangat penting, karena kebanyakan sistem merupakan bagian terlama dan

paling mahal dari proses pengembangan. Fase implementasi ini memiliki tiga langkah:

- a. Langkah awal adalah Konstruksi Sistem (*System Construction*). Sistem ini dibangun dan diuji untuk memastikan kinerja dari sistem, bekerja sebagaimana yang telah dirancang. Sebagian besar organisasi memberi lebih banyak waktu dan perhatian untuk menguji daripada mengubah program di tempat penerapan pertama.
- b. Instalasi Sistem, instalasi adalah proses dimana sistem lama dimatikan dan yang baru dinyalakan atau dihidupkan.
- c. Tim analis menetapkan sebuah rencana dukungan (*support plan*) untuk sistem. Rencana ini biasanya mencakup tinjauan pasca-pelaksanaan formal atau informal serta cara sistematis untuk mengidentifikasi perubahan besar dan kecil yang diperlukan untuk sistem tersebut.

2.5 Model *Waterfall*

Metode *Waterfall* adalah sebuah metode pengembangan sistem dimana antar satu fase ke fase yang lain dilakukan secara berurutan. Terdapat 5 langkah tahapan dalam metode *waterfall* adalah sebagai berikut:



Gambar II.3 Metode *Waterfall*

(Sumber: Alan Dennis, 2010)

Adapun penjelasan dari tahapan-tahapan Metode *Waterfall* menurut Dennis et al (2010) tersebut sebagai berikut:

1. Perencanaan (*Planning*)

Dalam tahapan ini, menjelaskan dan mengargumentasikan untuk melanjutkan proyek yang telah dipilih, rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Pada tahap ini ditentukan secara detail rencana kerja yang harus dikerjakan, durasi yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras, maupun financial diestimasi. Pembuatan perencanaan ini bukan langkah mudah karena untuk mengestimasi beban kerja dan durasi dari masing-masing tahap dibutuhkan pengalaman cukup banyak. Kesalahan pada tahap ini akan mengakibatkan keuntungan yang diperoleh tidak maksimal, bahkan bisa rugi. Pada tahapan ini peran manajemen sistem informasi berpengalaman yang dibutuhkan.

2. Analisis (*Analysis*)

Tahap kedua, adalah tahap analisis, yaitu tahap dimana kita berusaha mengenali segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan komponen-komponen sistem. Tujuan utama dari tahap analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Menganalisa kebutuhan sebagai bahan dalam membuat spesifikasi di tahapan selanjutnya.

3. Perancangan (*Design*)

Tahap perancangan (*design*) dimana kita mencoba mencari solusi permasalahan yang didapat dari tahap analisis. Tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang nyata untuk diimplementasikan. Jika pada tahapan analisis (*form requirement to specification*), maka tahapan desain adalah (*form specification to implementation*). Jadi, bagaimana pembuatan spesifikasi yang detail untuk bisa diimplementasikan.

4. Implementasi (*Implementation*)

Tahap implementasi, dimana kita mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Disini kita mulai berurusan dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (pengkodean/*coding*).

5. Sistem

Pada tahapan sistem dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem/perangkat lunak yang kita buat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat *iterative*, yaitu kembali ke tahap sebelumnya. Tahap pemeliharaan dan perawatan dimana kita mulai melakukan pengoperasian sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil. Kemudian jika waktu pengguna sistem habis, maka kita akan masuk lagi pada tahap perencanaan (*design*).

2.6 Analisis PIECES

Menurut Wukil Ragil (2010), metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Analisis ini disebut dengan *PIECES Analysis (Performance, Information, Economy, Control, Efficiency, and Service)*.

Analisis PIECES ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi, yaitu:

1. *Performance* (kinerja)

Memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan

seberapa handal suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan.

2. *Information* (informasi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan harus benar-benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan:

- a. Keluaran (*output*): Suatu sistem dalam memproduksi keluaran.
- b. Masukan (*input*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economics* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

4. *Control* (pengendalian)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan / kecurangan menjadi semakin baik pula.

5. *Efficiency* (efisiensi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem *manual*.

6. *Service* (layanan)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat *user friendly* untuk *user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

2.7 *System Requirements*

System requirements atau kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus

dicapai sistem baru (Satzinger, 2012). *System requirements* dibagi menjadi 2 kategori, yaitu:

1. *Functional Requirement*

Merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Selain itu kadang-kadang juga secara jelas menentukan apa yang tidak dikerjakan oleh sistem. *Functional Requirement* menggambarkan *system requirement* secara detil seperti input, output dan pengecualian yang berlaku. Contoh: Sistem mampu menginput dan menampilkan data pelamar kerja

2. *Non-Functional Requirement*

Secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Termasuk di dalamnya adalah batasan waktu, batasan proses pembangunan, standar-standar tertentu. Karena berkaitan dengan kebutuhan sistem secara keseluruhan, maka kegagalan memenuhi kebutuhan jenis ini berakibat pada sistem secara keseluruhan. Contoh: Hak akses admin seleksi penerimaan pegawai dapat melakukan semua kegiatan pada sistem sementara hak akses untuk kepala departemen hanya bisa melihat hasil akhir seleksi.

2.8 *Flowmap*

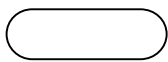

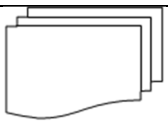
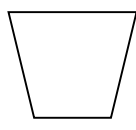
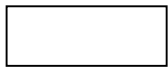
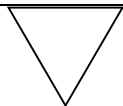
Flowmap atau bagan alir adalah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:

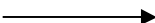

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukkan dengan jelas.

3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada Tabel II.1 berikut ini:

Tabel II.1 Simbol-simbol *Flowmap*

Simbol	Nama	Fungsi
	Mulai/Akhir	Menunjukkan dimulai/akhir dari sebuah proses.
	Dokumen	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>ouput</i> dicetak dikertas.
	Dokumen Rangkap	Menunjukkan multi dokumen.
	Manual <i>Process</i>	Simbol untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer/pekerjaan manual.
	Proses	Proses pengolahan data.
	Arsip	Menunjukkan pengarsipan data.

	Garis Alir	Arus dari suatu proses.
	<i>Input/Output</i>	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .

(Sumber: Jogiyanto, 2005)

2.9 Unified Modelling Language(UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Tujuan dari UML adalah untuk menyediakan kosa kata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain.

Menurut Dennis et al (2010), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure diagram* dan *Behavior diagram*. *Structure diagram* biasanya digunakan untuk merepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

Tabel II.2 Jenis Diagram Pada UML dan Fungsi-fungsinya

Nama Diagram	Fungsi
<i>Class Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan <i>class</i> di dalam sistem
<i>Object Diagram</i>	Menggambarkan hubungan antara pemodelan-pemodelan objek di dalam sistem
<i>Package Diagram</i>	Mengelompokkan elemen UML lainnya menjadi satu untuk membentuk level konstruksi yang lebih tinggi
<i>Deployment Diagram</i>	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen <i>software</i>

	sedang dijadikan arsitektur fisik suatu sistem
<i>Component Diagram</i>	Menggambarkan hubungan fisik diantara komponen-komponen <i>software</i>
<i>Composite Diagram</i>	Menggambarkan struktur internal dari suatu <i>class</i> dan hubungan diantara bagian-bagian dari suatu <i>class</i>

(Sumber: Dennis et al, 2010)

Tabel II.2 Jenis Diagram Pada UML dan Fungsi-fungsinya (Lanjutan)




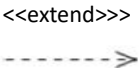
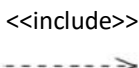


<i>Activity Diagram</i>	Menggambarkan proses bisnis masing-masing <i>class</i>
<i>Sequence Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam aktifitas berdasarkan urutan waktu
<i>Communication Diagram</i>	Memodelkan kebiasaan dari objek-objek dalam suatu <i>use case</i> . Fokus dalam komunikasi antara satu set dari kolaborasi objek dari suatu objek
<i>Interaction Overview Diagram</i>	Menggambarkan suatu ikhtisar alur dari kontrol suatu proses
<i>Timing Diagram</i>	Menggambarkan interaksi yang terjadi diantara suatu set dari objek-objek dan perubahan keadaan selama perjalanan waktu
<i>Behavioral State Machine Diagram</i>	Memeriksa kebiasaan dari suatu <i>class</i>
<i>Protocol State Machine Diagram</i>	Menggambarkan ketergantungan diantara perbedaan-perbedaan <i>interface</i> dari suatu <i>class</i>
<i>Use Case Diagram</i>	Menangkap kebutuhan bisnis untuk sistem dan untuk menggambarkan interaksi diantara sistem dan lingkungannya

(Sumber: Dennis et al, 2010)

2.9.1 Use Case Diagram

Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah apa yang dapat diperbuat oleh sistem, dan bukan bagaimana sistem melakukannya. Diagram *use case* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Simbol-simbol yang terdapat didalam *use case diagram* dijelaskan pada tabel II.3 dibawah ini:

Tabel II.3 Simbol-simbol *Use case Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Use case</i>	Merepresentasikan bagian utama dari sistem secara fungsional.
2		<i>Actor</i>	Seseorang atau sistem yang mendapatkkn keuntungan dari sistem
3		<i>Association</i>	Menghubungkan suatu <i>actor</i> dengan <i>use case</i> .
4		<i>Extend</i>	Merepresentasikan eksistensi dari <i>use case</i> untuk menyertakan perilaku <i>optional</i> .
5		<i>Include</i>	Mewakili dimasukkannya satu <i>use case</i> ke <i>use case</i> yang lain.
6		<i>Subject</i>	Menyertakan nama subjek didalam maupun diatas
7		<i>Generalization</i>	Merepresentasikan <i>use case</i> khusus ke yang lebih umum.

(Sumber: Dennis, 2015)

2.9.2 Use Case Description

Use Case Description merupakan deskripsi yang mencatat mengenai detail pemrosesan dari suatu *use case* (Satzinger et al, 2010). *Use Case Description* memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan suatu proses bisnis. Menurut Dennis et al (2010:168), elemen yang dimiliki *Use Case Description* adalah sebagai berikut:

1. Basic Information

a. Nama

Nama digunakan untuk menjelaskan *use case* dan membedakan *use case* satu dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi yang memungkinkan.

b. Description

Description berarti uraian singkat untuk menyampaikan tujuan dari *use case*.

c. Aktor

Aktor merujuk kepada orang, sistem, perangkat lunak ataupun perangkat keras yang berinteraksi langsung dengan sistem untuk mencapai tujuan.

2. Normal Courses

Bagian utama dari *use case* berikutnya adalah *normal courses*. Pada *normal courses* menjelaskan mengenai langkah-langkah yang dilakukan untuk mengeksekusi respon terhadap masukan dan keluaran.

3. Alternative Courses

Untuk beberapa *use case* memiliki jalur alternatif pada setiap kasusnya. Pada penggambaran jalur alternatif ini terdapat cabang yang juga akan menghasilkan kesimpulan dari *use case* tersebut.

2.9.3 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku dalam proses bisnis yang independen terhadap objek. Dalam banyak hal, *activity diagram* dapat


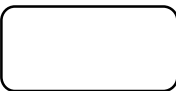
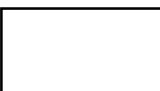
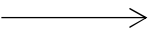

dipandang sebagai data flow diagram canggih yang digunakan pada konjungsi dengan analisis terstruktur. Walaupun begitu, *activity diagram* tidak seperti data *flow diagram*, *activity diagram* menggunakan notasi yang mengalamatkan pemodelan paralel. (Dennis, 2010)



Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan actor, jadi aktifitas yang dapat dilakukan oleh sistem. Diagram aktifitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang akan ditampilkan pada perangkat lunak

Berikut adalah simbol-simbol yang ada pada Activity Diagram:

Tabel II.4 Simbol-simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1		<i>Action</i>	Merupakan notasi yang simpel, dan tidak untuk dianalisis perilakunya. Dinamakan sesuai aksi yg dilakukan.
2		<i>Activity</i>	Digunakan untuk mewakili satu set aksi yang dilakukan.
3		<i>Object Node</i>	Digunakan untuk mewakili objek yang terhubung ke satu set arus objek.
4		<i>Control Flow</i>	Menampilkan urutan eksekusi.
5		<i>Object Flow</i>	Menunjukkan aliran dari sebuah objek dari satu aktifitas ke aktifitas lainnya.

6		<i>Initial Node</i>	Menggambarkan permulaan dari suatu set aktifitas atau aksi.
7		<i>Final-activity Node</i>	Menggambarkan akhir dari setiap aktifitas.



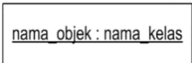
(Sumber: Dennis, 2015)


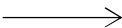

2.9.4 Sequence Diagram

Sequence diagram adalah salah satu dari dua jenis diagram interaksi. *Sequence diagram* menggambarkan benda-benda yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*. *Sequence Diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi didefinisikan.

Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set benda, diagram tersebut sangat membantu untuk memahami spesifikasi *real-time* dan kompleks menggunakan kasus (Dennis, 2011). Berikut simbol yang ada pada *Sequence Diagram*:

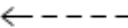
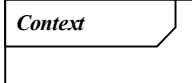
Tabel II.5 Simbol-simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1.		Aktor	Orang atau sistem yang berasal dari manfaat dan eksternal ke sistem yang berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan
2.		Garis Hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Berpartisipasi secara berurutan dengan mengirim dan atau menerima pesan yang ditempatkan diatas diagram.

4.		<i>Execution Occurrence</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5.		<i>Message</i>	Pesan yang menggambarkan komunikasi yang terjadi antar objek.
6.		<i>Message (return)</i>	Pesan yang dikirim untuk diri sendiri secara langsung.

(Sumber: Dennis, 2011)

Tabel II.5 Simbol-simbol *Sequence Diagram* (Lanjutan)

No	Simbol	Nama	Deskripsi
7.		<i>Message (return)</i>	Pesan yang dikirim untuk diri sendiri.
8.	X	<i>Object Destruction</i>	Sebuah x ditempatkan di ujung suatu objek garis hidup untuk menunjukkan bahwa akan keluar dari eksistensi.
9.		<i>Frame</i>	Menunjukkan konteks dari <i>sequence diagram</i> .


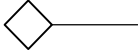
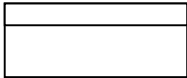
(Sumber: Alan Dennis, 2011)

2.9.5 Class Diagram

Sebuah diagram kelas adalah model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda /fungsi) (Alan Dennis, 2011). Berikut simbol-simbol yang ada pada diagram kelas:

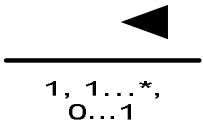
Tabel II.5 Simbol-simbol *Class Diagram*

No	Simbol	Nama	Deskripsi
----	--------	------	-----------

1		<i>Generalization</i>	Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.
2		<i>Aggregation</i>	Menggambarkan suatu <i>class</i> terdiri dari <i>class</i> lain atau suatu <i>class</i> adalah bagian dari <i>class</i> lain.
3		<i>Class</i>	Kelas pada struktur sistem

(Sumber: Alan Dennis, 2011)

Tabel II.5 Simbol-simbol *Class Diagram* (Lanjutan)

No	Simbol	Nama	Deskripsi
4		<i>Association</i>	Asosiasi yang menghubungkan <i>class</i> dengan <i>class Multiplicity</i> .

(Sumber: Alan Dennis, 2011)

2.10 Database




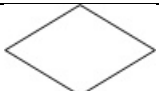
Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/ berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Fathansyah, 2012). Sebagai satu kesatuan istilah, Basis data (Database) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file/table* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.11 Entity Relationship Diagram(ERD)

Terdapat sebuah model perancangan hubungan antar entitas (tabel) dari sebuah basis data (*database*). Istilah untuk frase ini biasa dikenal dengan nama *Entity Relationship Model*. Model hubungan ini seterusnya akan berlanjut menjadi sebuah Diagram Hubungan Antar Entitas yang biasa dikenal dengan nama *Entity Relationship Diagram (ERD)* (Jeffrey A. Hoffer, 2007). Simbol-simbol ERD dapat dilihat pada Tabel II.8.

Tabel II.8 Simbol-Simbol ERD

Simbol	Nama	Deskripsi
	<i>Strong Entity</i>	Setiap hal dunia nyata (orang, tempat, objek, konsep, aktivitas) tentang suatu perusahaan mencatat data
	<i>Attribute</i>	Properti atau karakteristik tipe entitas.
	<i>Multivalued Attribute</i>	Karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relationship</i>	Hubungan antar entitas.

(Sumber: Jeffrey A. Hoffer, 2007)

2.12 Kamus Data

Menurut Jogiyanto (2005), kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data diharapkan, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan di dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Keuntungan penggunaan kamus data adalah (Sommerville, 2003):

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Kamus data dapat memeriksa keunikan nama dan memberitahu analis persyaratan sekiranya terjadi duplikasi nama.
2. Kamus data sebagai tempat penyimpanan informasi yang dapat menghubungkan analis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat. Berikut adalah contoh penulisan kamus data:

Spesifikasi Tabel pemasok

Nama Tabel : Pemasok

Tipe : *File master*

Tabel II.7 Contoh Kamus Data Untuk Tabel Pemasok

No	Nama Elemen	Akronim	Tipe	Width	Keterangan
1.	ID pemasok	ID_pemasok	Varchar	10	<i>Primary Key</i>

2.	Nama pemesok	Nama_pemasok	Char	40	
3.	Alamat pemesok	Alamat	Varchar	100	
4.	Nomor telepon	Telepon	Varchar	12	

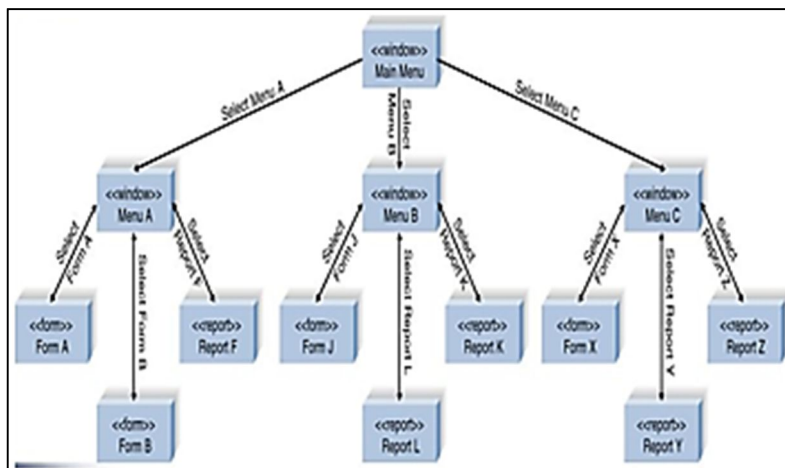
(Sumber: Jogiyanto, 2005)

2.13 *Windows Navigation Diagram*(WND)

Windows Navigation Diagram merupakan *statechart* diagram khusus yang berfokus pada *user interface*. *Windows Navigation Diagram* (WND) menurut Dennis (2010) adalah sebagai berikut:

1. Menunjukkan bagaimana semua layar, formulir, dan laporan terkait.
2. Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
3. Seperti diagram keadaan untuk *user interface*.
 - a. Kotak mewakili komponen.
 - b. Panah mewakili transisi.
 - c. Stereotipe menunjukkan tipe antarmuka.

Contoh dari *Windows Navigation Diagram* dapat dilihat pada Gambar II.4.



Gambar II.4 Contoh *Windows Navigation Diagram*

(Sumber: Dennis, 2010)

2.14 *Hypertext Preprocessor* (PHP)

Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang digunakan untuk aplikasi web yang dinamis dan interaktif. Sebuah halaman PHP adalah sebuah halaman HTML yang memiliki *server-side scripts* yang ditempatkan dalam *server* dan diproses oleh web *server* sebelum dikirim ke *browser* pemakai (Welling dan Thomson, 2003).

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh web *server*, dimana proses script perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui web *browser* berupa tampilan HTML. Menurut Welling dan Thomson (2003), beberapa keunggulan PHP adalah:

1. *High Performance*
PHP sangat efisien. Dengan menggunakan *server* tunggal yang tidak mahal, *user* dapat melakukan banyak pekerjaan setiap harinya.
2. *Database Integration*
PHP mempunyai sambungan ke banyak sistem basis data, antara lain MySQL, PostgreSQL, Oracle, Informix, dan Sysbase Databases.
3. *Built-in-Libraries*
PHP dirancang khusus untuk web, dan mempunyai banyak *built-in-function* untuk menampilkan banyak fungsi di dalam web.
4. Harga yang murah
PHP adalah perangkat lunak gratis.
5. Mudah dalam pembelajaran dan penggunaan
Sintaks PHP berdasarkan bahasa pemrograman lainnya, terutama C dan Java.
6. *Portability*
PHP dapat digunakan di banyak sistem operasi yang berbeda.
7. Ketersediaan *Source Code*
Kode PHP dapat langsung diakses dan dimodifikasi secara bebas.

2.15 XAMPP

XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolahan data MySQL di komputer lokal. XAMPP berperan sebagai *web server* pada komputer. XAMPP juga dapat disebut sebagai sebuah *CPanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses internet (Kadir, 2014).

Fungsi lainnya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X, Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang tabel (Sutaji,2012).

2.16 MySQL

My Structured Query Language (MySQL) adalah *Relational Database Management System* (RDBMS) yang sangat cepat dan kuat. Sebuah basis data memungkinkan user untuk melakukan penyimpanan yang efisien, pencarian dan pengurutan data. MySQL *server* memiliki kemampuan melakukan kontrol akses terhadap data untuk memastikan bahwa setiap *user* dapat bekerja dengan sesuai, menyediakan akses yang cepat, dan meyakinkan bahwa hanya *user* yang mempunyai hak akses yang dapat mengaksesnya. Oleh karena itu, MySQL merupakan *database server* yang yang dapat digunakan banyak *user* dan banyak pekerjaan.

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structure Query Language*) (Sutaji, 2012).

2.16.1 Tipe data MySQL

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang *table* (Sutaji, 2012).

Jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II.9 berikut:

Tabel II.9 Tipe Data MySQL

Tipe Data	Keterangan
<i>CHAR</i>	Sebuah <i>string</i> dengan panjang tetap. Sisa Jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter.
<i>VARCHAR</i>	<i>String</i> dengan panjang berupa variabel. M bisa mencapai 65535.
<i>DATE</i>	Data berupa tanggal. Format tanggal dalam bentuk ‘YYYY-MM-DD’.
<i>TIME</i>	Data berupa waktu. Format waktu dalam bentuk ‘HH:MM:SS’.
<i>TINYINT</i>	Bilangan antara -128 sampai dengan +127.
<i>SMALLINT</i>	Bilangan antara -32768 sampai dengan +32767.

(Sumber: Sutaji, 2012)

Tabel II.9 Tipe Data MySQL (Lanjutan)

Tipe Data	Keterangan
<i>MEDIUMINT</i>	Bilangan antara -8388608 sampai dengan +8388607.
<i>INT</i>	Bilangan antara -2147683648 sampai dengan +2147683647
<i>FLOAT</i>	Bilangan <i>floating point</i> yang kecil (presisi tunggal). Jangkauan

	nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38.
<i>DOUBLE</i>	Bilangan <i>floating point</i> dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan 2.225073858507201E-308 s/d 1.7976931348623157E+308.
<i>ENUM</i>	Sebuah <i>enumeration</i> . Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1', 'value2', ..., NULL atau nilai spesial "" <i>error</i> . Sebuah <i>enum</i> dapat menampung 65535 pilihan nilai.

(Sumber: Sutaji, 2012)

2.17 *Blackbox Testing*

Black-Box Testing adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur internal atau cara kerjanya. Metode pengujian ini dapat diterapkan secara virtual untuk setiap tingkat pengujian perangkat lunak (unit, integrasi, sistem, dan penerimaan). Kadang-kadang disebut sebagai pengujian berbasis spesifikasi (Ron Patton, 2005).

Contoh kegiatan yang dilakukan pada *black-box testing* yaitu:

1. Membuat *test case* untuk menguji fungsi-fungsi yang ada pada aplikasi
2. Membuat *test case* untuk menguji kesesuaian alur kerja suatu fungsi di aplikasi dengan *requirement* yang dibutuhkan *user* untuk fungsi tersebut.
3. Mencari *bugs/error* dari tampilan (*Interface*) aplikasi.

BAB III

METODOLOGI PENELITIAN

1.7 Metodologi Penelitian

Metodologi penelitian merupakan proses ataupun cara ilmiah yang digunakan untuk mendapatkan data yang akan digunakan untuk keperluan penelitian. Metodologi penelitian juga sering dikenal sebagai metode ilmiah, metode tersebut digunakan untuk mencari, mengembangkan, serta menguji suatu kebenaran pengetahuan. Metodologi penelitian mempelajari cara-cara melakukan pengamatan dengan pemikiran yang tepat secara terpadu melalui tahapan-tahapan yang disusun secara ilmiah untuk mencari, menyusun, serta menganalisis dan menyimpulkan data-data berdasarkan fakta-fakta secara ilmiah.

1.8 Pengumpulan Data

Terdapat 2 jenis data yang diperlukan dalam melakukan pengumpulan data, yaitu data primer dan data sekunder. Sumber dari data-data ini diperoleh dari tempat penulis melakukan praktek kerja lapangan yang dilakukan di PT Krama Yudha Ratu Motor. Adapun jenis data yang dikumpulkan meliputi:

1. Data Primer

Data primer merupakan data yang diperoleh secara langsung dari perusahaan, dimana pengumpulan data atau informasi dilakukan langsung diperoleh dari objek yang diteliti. Data tersebut data yang digunakan dalam berlangsungnya proses penjadwalan produksi Mitsubishi *Colt* T120ss, diantaranya struktur organisasi, SOP, proses bisnis yang sedang berjalan saat ini, dan dokumen-dokumen yang terlibat dalam proses penjadwalan produksi yang sedang berjalan.

2. Data Sekunder

Data sekunder merupakan data yang diperoleh dari data yang tersedia, dan yang sudah ada terlebih dahulu dikumpulkan dan dilaporkan kepada pihak lain, buku-buku, dan kajian ilmiah dari berbagai sumber yang berkaitan

dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan, serta struktur organisasi perusahaan

1.9 Metode Pengumpulan Data

Menurut Sugiyono (2013), teknik pengumpulan data merupakan langkah yang paling strategis dalam melakukan penelitian, karena tujuan utama dari penelitian tersebut adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

1. Observasi

Observasi yaitu melakukan pengamatan situasi langsung pada sistem yang berjalan di Divisi *Product Planning & Inventory Control* pada PT Krama Yudha Ratu Motor. Hasil pengamatan yang dilakukan menjadi landasan dalam melakukan pengembangan sistem yang akan dibuat.

2. Wawancara

Metode wawancara adalah metode pengumpulan data melalui tanya jawab secara langsung kepada pihak-pihak yang terkait dengan objek penelitian. Pihak yang diwawancarai adalah *pegawaif* PPC dan *Foreman* PPC di Divisi PPIC.

3. Studi Pustaka

Studi pustaka yaitu metode pengumpulan data dengan melakukan pencarian referensi yang berkaitan dengan pemrograman PHP dan *database* MySQL dari berbagai referensi, baik itu referensi elektronik yang didapat dari internet maupun referensi dari buku teks. Referensi yang diperoleh, kemudian dikaji sebagai dasar dalam menyelesaikan penelitian.

1.10 Metode Pengembangan Sistem

Menurut Dennis (2015) *System Development Life Cycle* (SDLC) memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa cara dalam membangun sebuah sistem mungkin dapat mempunyai perbedaan, tetapi hampir semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun dari beberapa langkah-langkah yang

menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik dan file yang menjelaskan pemahaman tentang proyek. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *waterfall*.

Metode *waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Metode *waterfall* menyediakan pendekatan alur hidup perangkat lunak dengan terurut dari fase awal (*planning*) hingga fase akhir (*implementation*), dan pada setiap fase pihak pengembang dan sponsor akan selalu ada kontak untuk mendapatkan persetujuan sebelum masuk pada tahap selanjutnya (Dennis, 2010).

Tahapan-tahapan dalam *waterfall* adalah sebagai berikut:

- a. Identifikasi kebutuhan pengguna (*planning*)
Pengembang melakukan diskusi dengan *manager* divisi *Product Planning & Inventory Control* tentang kebutuhan sistem yang digunakan.
- b. Analisis kebutuhan perangkat lunak (*Analysis*)
Pengembang melakukan proses pengumpulan kebutuhan yang dilakukan secara intensif dengan pegawai divisi *Product Planning & Inventory Control* tentang kebutuhan sistem yang diinginkan.
- c. Desain (*Design*)
Pengembang membuat sebuah desain program perangkat lunak seperti struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean serta dokumentasinya dari sistem yang telah dijelaskan oleh *Foreman* dan pegawai divisi *Product Planning & Inventory Control*
- d. Pembuatan Kode Program (*Implementaion*)
Pengembang mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam suatu bahasa pemrograman yang dapat dibaca oleh mesin.
- e. Sistem (*System*)
Pada tahap ini pengembang melakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem atau perangkat lunak yang sudah dibuat sesuai dengan kebutuhan pengguna atau belum

1.11 Literature Review

Literature review merupakan evaluasi yang terperinci serta kritis tentang penelitian sebelumnya pada suatu topik yang sedang dibicarakan. Termasuk peran penting dalam membuat sebuah karya ilmiah, maupun tulisan karena dapat memberikan ide dan tujuan tentang topik penelitian yang akan dilaksanakan. Pada umumnya, *Literature review* memiliki isi meliputi ulasan, rangkuman, dan pemikiran penulis tentang beberapa pustaka. Adapun beberapa langkah dalam melakukan *literature review* yaitu:

1. Formulasi Permasalahan

Penulis memilih topik yang sesuai, selain itu permasalahan yang diangkat harus sesuai, dalam hal ini sesuai dengan penelitian yang akan dilakukan

2. Mencari Literatur

Literatur harus relevan dengan penelitian yang sedang dilakukan, sehingga dapat membantu untuk mendapatkan gambaran dari suatu topik dan objek penelitian

3. Evaluasi Data

Membandingkan setiap literatur yang ada, lalu mencari kesamaan, maupun perbedaan dengan topik yang akan dibahas.

Berikut merupakan *literature review* dari beberapa jurnal ilmiah yang dapat dilihat pada Tabel III.1

Tabel III. 1 *Literature Review*

No	Pengarang/Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Ulasan
1	Riskiyadi Supriyadi, 2016	Penjadwalan Produksi Iks-Filler	Belum tersedianya jadwal induk produksi sebagai dasar penentuan proses produksi, sehingga terjadi kekurangan persediaan	<i>Master Production Schedule (MPS)</i>	Penerapan metode Master Production Schedule (MPS) menghasilkan analisis dan penyelesaian masalah yang terjadi pada perusahaan tersebut	Persamaan: Di dalam jurnal penelitian penjadwalan produksi Iks-Filler pada proses <i>Ground Calcium Carbonate</i> di perusahaan kertas memiliki beberapa kesamaan dalam pembahasan dengan tugas akhir yang dibuat oleh penulis, salah satunya adalah metode yang digunakan yaitu metode (<i>Master Production Schedule</i>). Adapun tujuan utama jurnal memiliki kesamaan, yaitu

						<p>meminimalisir terjadinya kesalahan dalam penjadwalan produksi.</p> <p>Perbedaan:</p> <p>Adapun perbedaan yang ada di dalam jurnal Riskiyadi Supriyadi dengan Tugas Akhir. Tugas Akhir yang dibuat oleh penulis berupa suatu sistem informasi penjadwalan produksi berbasis komputer</p>
--	--	--	--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabel III. 1 *Literature Review* (Lanjutan)

2	Iwan Hartono, 2014	Aplikasi Penjadwalan Produksi Manufaktur berbasis <i>Web</i>	Kendala dalam menentukan kegiatan produksi harus dilakukan.	<i>Master Production Schedule (MPS)</i>	Penerapan aplikasi penjadwalan produksi berbasis <i>web</i> dapat merencanakan kegiatan produksi, sehingga dapat menyelesaikan masalah pada jurnal tersebut	<p>Persamaan: Di dalam jurnal penelitian aplikasi penjadwalan produksi manufaktur berbasis web oleh Iwan Hartono, terdapat beberapa persamaan dengan tugas akhir penulis. Salah satunya adalah tujuan utama, yaitu membuat aplikasi penjadwalan produksi, serta metode yang digunakan adalah <i>Master Production Schedule</i>, dimana metode yang digunakan sama dengan tugas akhir penulis</p> <p>Perbedaan: Di dalam jurnal penelitian</p>
---	--------------------	--------------------------------------------------------------	-------------------------------------------------------------	-----------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

						<p>aplikasi penjadwalan produksi ini memiliki perbedaan dengan tugas akhir penulis, yaitu berupa antar muka pengguna, <i>database</i>, serta data yang dikumpulkan tidak sama. Aplikasi yang dibuat oleh Iwan Hartono memiliki fungsi secara universal, sementara penulis membuat lebih spesifik untuk PT Krama Yudha Ratu Motor</p>
--	--	--	--	--	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sumber: Hasil Ulasan Penulis, (2018)

1.12 Kerangka Pemecahan Masalah

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam pembuatan program yang ada sesuai dengan metodologi penelitian yang digunakan dalam pembuatan program tersebut. Penjelasan langkah-langkah atau tahapan dalam pembuatan program tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Pada tahap awal penulis melakukan sebuah studi pendahuluan yaitu dengan membaca buku literatur, *browsing internet*, membaca artikel serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan . semua yang berhubungan dengan judul dan permasalahan tugas akhir yang di ambil dibaca oleh penulis. Selain itu, penulis juga melakukan proses pemagangan yang dilakukan di PT Krama Yudha Ratu Motor pada divisi *Production Planning & Inventory Control* dalam jangka waktu 1 bulan penuh

2. Identifikasi Masalah

Pada tahap ini dilakukan cara dalam mengidentifikasi suatu masalah yang terjadi pada divisi *Production Planning & Inventory Control* yang berfokus pada penjadwalan produksi dalam mengidentifikasi suatu masalah penulis melakukan beberapa cara seperti:

- a. Melakukan wawancara dan observasi yang melibatkan PT Krama Yudha Ratu Motor dan beberapa karyawan yang bekerja pada bagian PPC sebagai sumbernya
- b. Mengumpulkan data-data proses bisnis yang berjalan,
- c. Menggunakan analisis PIECES untuk identifikasi permasalahan.

3. Identifikasi Pemecahan Masalah

Setelah mengidentifikasi suatu masalah, maka penulis melakukan pemecahan suatu masalah berdasarkan data-data yang telah di dapat pada tahap sebelumnya. Terdapat beberapa pemecahan masalah yang penulis lakukan seperti:

- a. Menentukan tujuan penelitian serta batasan-batasan pada tugas akhir yang ditulis oleh penulis
- b. Menerapkan metode *Master Production Schedule* (MPS) untuk menjadwalkan produksi sesuai dengan teknik yang ada dalam penjadwalan produksi
- c. Mengembangkan sistem dengan menggunakan metode *waterfall*, dengan alasan dokumen yang akan diolah akan terorganisir dengan baik. Hal ini dikarenakan setiap fase yang dilalui harus selesai semua terlebih dahulu sebelum masuk ke fase berikutnya.

4. Identifikasi Kebutuhan Pengguna (*Analysis*)

Pada tahap ini penulis melakukan identifikasi kebutuhan sistem dengan beberapa cara seperti:

- a. Mengumpulkan data-data pada proses bisnis yang berjalan, pengumpulan data-data permasalahan serta menganalisis dokumen masuk dan keluar pada proses penjadwalan produksi.
- b. Identifikasi proses bisnis yang berjalan lalu menganalisis proses bisnis tersebut.
- c. Identifikasi kebutuhan sistem informasi penjadwalan produksi.

5. Desain

Pada tahap ini penulis melakukan perancangan dalam pembuatan sistem dengan menggunakan UML diagram yang bertujuan untuk memodelkan sistem. Dalam pembuatan UML terdapat beberapa diagram seperti:

- a. *Use case Diagram* yang bertujuan untuk mendeskripsikan sebuah interaksi antara *user* (aktor) dengan sistem yang akan di rancang.
- b. *Use case Description* yang bertujuan untuk menjelaskan setiap *use case* yang telah dibuat secara lebih spesifik
- c. *Activity Diagram* yang bertujuan untuk menggambarkan *workflow* atau aliran kerja pada proses bisnis.
- d. *Class Diagram* yang bertujuan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat di antara objek-objek tersebut.

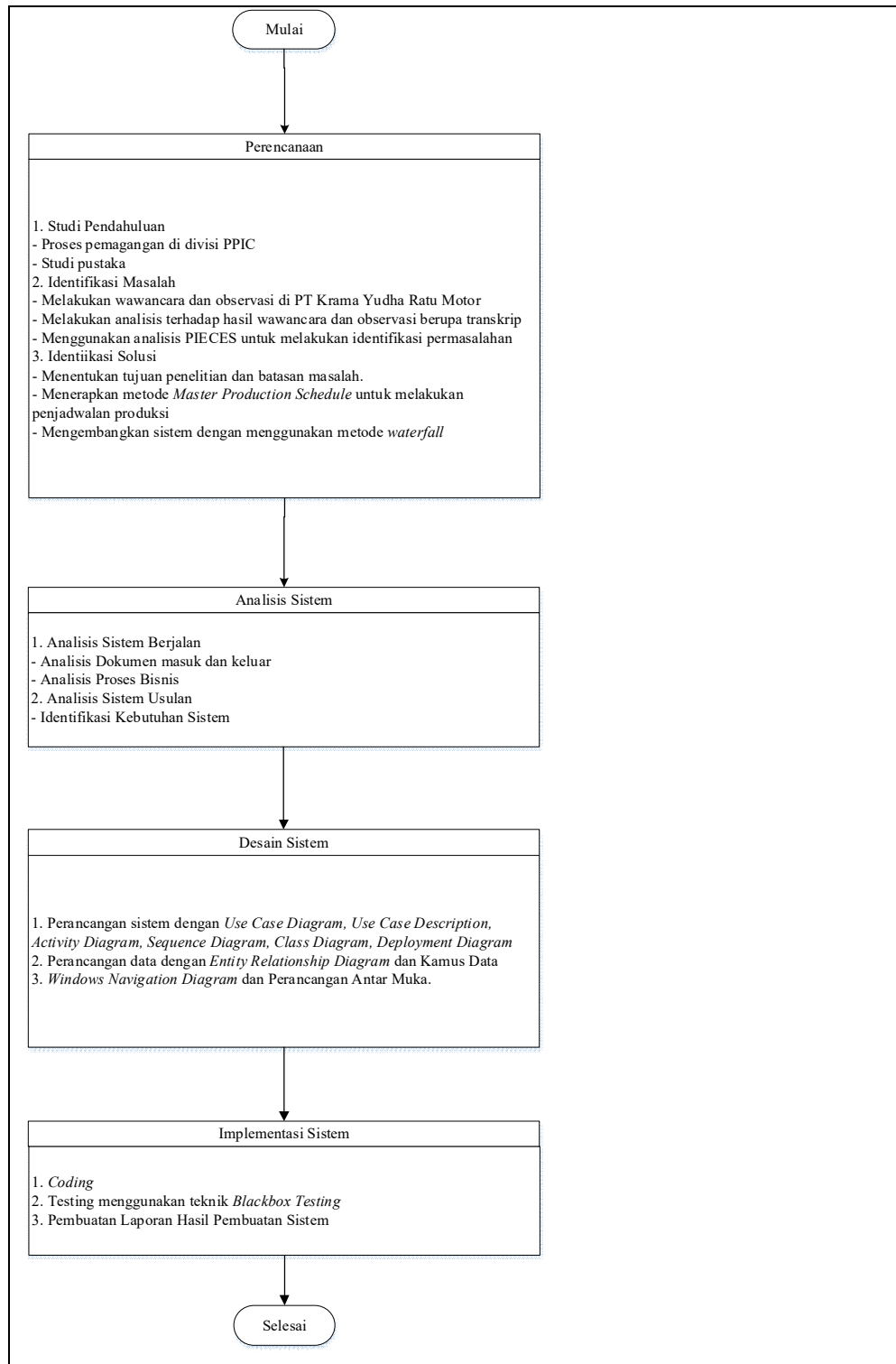
- e. *Sequence Diagram* bertujuan untuk menggambarkan kelakuan atau perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang diterima dan dikeluarkan antar objek.
- f. *Deployment Diagram* yang bertujuan untuk memvisualisasikan, menspesifikasikan dan mendokumentasi proses yang terjadi pada suatu sistem perangkat lunak yang akan dibangun.

Selain itu dilakukan pembuatan *User Interface* dengan metode *windows Navigation Diagram* (WND) yang bertujuan dalam memberikan suatu gambaran navigasi yang terdapat dalam sistem yang akan dibuat, pembuatan *Entity Relationship Diagram* (ERD) yang bertujuan untuk memodelkan suatu data dan hubungan data serta pembuatan kamus data dengan tujuan untuk memvalidasi diagram alir data dalam hal kelengkapan dan keakuratan data.

6. Implementasi Sistem

Pada tahap terakhir ini penulis melakukan penulisan kode program dan pembuatan *database* menggunakan PHP versi 5.6.28 dan MySQL versi 5.0.11 yang semuanya dijalankan melalui XAMPP v3.2.2. sesuai dengan analisis yang telah dilakukan pada tahap sebelumnya, selain itu dilakukan sebuah pengujian dengan menggunakan *blackbox testing* yang bertujuan untuk menguji coba apakah setiap fungsi atau tombol-tombol yang terdapat pada sistem dapat berjalan sesuai yang direncanakan atau diharapkan oleh penulis dan setelah itu penulis melakukan pembuatan laporan pengembangan sistem informasi penyimpanan bahan baku.

Kerangka penelitian yang telah dibuat untuk laporan Tugas Akhir ini dapat dilihat pada Gambar III.1



Gambar III. 1 Kerangka Penelitian

Sumber: Hasil Penelitian(2018)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Tentang Perusahaan

PT Krama Yudha Ratu Motor didirikan pada tanggal 1 Juni 1973 sebagai perusahaan swasta dengan 100% modalnya merupakan Penanaman Modal Asing (PMA). Saat ini PT Krama Yudha Ratu Motor sudah mendapatkan sertifikat ISO 9001 sejak tahun 2002 untuk sistem prosedur yang digunakan dan mempunyai dokumen proses bisnis yaitu SSP (Standar Sistem Prosedur) dan Manual Mutu. Pendirian PT Krama Yudha Ratu Motor tersebut berdasarkan Akte Notaris Abdul Latief No. 16 tanggal 1 Juni tahun 1973. Diperkuat lagi dengan:

1. Perizinan dari BKPM dalam bidang usaha No. 92/A/BKPM/73/PMDN tanggal 4 Juli 1973.
2. Perizinan dari Departemen Perindustrian dalam bidang teknis No. 27/2A/D/IV/74 tanggal 21 Maret 1974, pada saat itu perusahaan ini masuk dalam kelompok *assembling*, mesin dan perbengkelan yang kini menjadi kelompok otomotif (beroda 4 atau lebih).
3. Perizinan dari Departemen Kehakiman dalam bidang hukum No. 16.A.S.105/18/74 tanggal 15 April 1974.
4. Perizinan dari Pengadilan Negeri Jakarta dalam bidang hukum kewilayahan No. 1374 tanggal 18 April 1974.

PT Krama Yudha Ratu Motor merupakan sebuah perseroan terbatas yang bergerak dalam bidang perakitan kendaraan bermotor jenis niaga. PT Krama Yudha Ratu Motor ini merupakan bagian dari Krama Yudha Mitsubishi Group (KYMG). Awal berdirinya KYMG adalah akibat banyaknya kendaraan bermotor dari Eropa yang diimpor ke Indonesia. Guna mengurangi pengimporan kendaraan tersebut, maka para pengusaha melakukan pertemuan dan bersepakat untuk mendirikan suatu perusahaan perakitan kendaraan bermotor di Indonesia dengan menggunakan lisensi dari Mitsubishi Motor Corporation Jepang.

Krama Yudha Mitsubishi Group terbagi atas PT Krama Yudha Holding yang berdiri pada tahun 1969 di Jakarta, yang kemudian menjadi induk dari beberapa perusahaan di bidang produksi kendaraan bermotor merk Mitsubishi. PT Krama Yudha Holding ini juga memiliki anak perusahaan di beberapa tempat yaitu:

1. PT Krama Yudha Ratu Motor yang merupakan pabrik perakitan kendaraan bermotor jenis niaga yang berdiri pada tanggal 1 Juni 1973.
2. PT Mitsubishi Krama Yudha Motor & Manufacturing I dan 2 didirikan pada tahun 1975 dan 1981. PT Mitsubishi Krama Yudha Manufacturing ini merupakan pabrik pembuatan komponen dan suku cadang kendaraan bermotor merk Mitsubishi yang dirakit di dalam negeri.
3. PT Krama Yudha Tiga Berlian berdiri pada tahun 1972 dan bertindak sebagai importir serta distributor tunggal kendaraan merk Mitsubishi.
4. PT Krama Yudha Mojopahit Motor didirikan pada tahun 1975 dan ditutup pada tahun 1986. Perusahaan ini merakit kendaraan bermotor merk Mitsubishi jenis *Colt Diesel* FE.101 dan *Colt Diesel* FE.114.
5. PT Krama Yudha Kesuma Motor didirikan pada tahun 1981 yang bergerak di bidang perakitan kendaraan bermotor jenis sedan dan penumpang serta sedan *Galant 2 Type 1400 Saloon* F 1400 *Hatch Back*.

Pada bulan Januari 1975 PT Krama Yudha Ratu Motor mulai merakit atau mulai menghasilkan produksi komersilnya dengan menggunakan peralatan dan tempat yang cukup baik. Dalam tahun tersebut perusahaan ini menghasilkan kendaraan bermotor jenis niaga berjumlah 4427 unit yang terdiri dari:

1. Kendaraan Niaga *Type T 120 Pick Up* sebanyak 383 unit.
2. Kendaraan Niaga *Type T210 CN* sebanyak 108 unit.
3. Kendaraan Niaga *Type 200 CU* sebanyak 606 unit.
4. Kendaraan Niaga *Type T 210 FZ* sebanyak 1772 unit.
5. Kendaraan Niaga *Type 633 E* sebanyak 1558 unit.

Tipe kendaraan yang dirakit di PT Krama Yudha Ratu Motor, yaitu:

1. *CJM Gas Pick Up*, *CJM Gas Mini Bus*
2. *SL Diesel Pick Up*, *SL Diesel Mini Bus*
3. *Colt Diesel* FE-304, FE-334, FE-347, FE-349, FE-447

4. *Fuso* FM 517.F, *Fuso* FM 517.H, *Fuso* FN-517, *Fuso* FN-527
5. *Outlander* GLS, GLX, PX

4.2 Profil Perusahaan

Krama Yudha Ratu Motor adalah perusahaan manufaktur otomotif yang berada di Cakung, Jakarta Timur. Berikut profil dari PT Krama Yudha Ratu Motor. Berikut merupakan pabrik PT Krama Yudha Ratu Motor pada Gambar IV.1.



Gambar IV.1. PT Krama Yudha Ratu Motor

Sumber: PT Krama Yudha Ratu Motor (2017)

Nama	: PT Krama Yudha Ratu Motor
Alamat	: Jl. Raya Bekasi KM, 21-22 Rawa Terate, Cakung, Pulo Gadung-Jakarta 13920 Telp. (021) 4602905 Fax (021) 4602904
Tanggal Didirikan	: 1 Juni 1973
Bangunan Pabrik	: 20.360 m ²
Luas Tanah	: 143.035 m ²
Bangunan Pendukung	: 6.600 m ²
Produksi Komersial	: Sejak Tahun 1975
Total Karyawan	: 1,468 (Agustus 2017)

Jenis Usaha : Perakitan Kendaraan Bermotor Merek Mitsubishi

Sementara itu, PT Krama Yudha Ratu Motor memiliki para tenaga kerja yang berjasa dalam pemenuhan kebutuhan kendaraan roda empat atau lebih di Indonesia. Oleh karena itu, untuk meningkatkan kedisiplinan semua tenaga kerja, maka PT Krama Yudha Ratu Motor selalu menekankan program budaya 5R, yaitu:

1. Ringkas: suatu usaha dalam memisahkan barang yang diperlukan dengan yang tidak diperlukan.
2. Rapi: usaha yang selalu memastikan segala sesuatu pada tempatnya dan juga pemberian label agar mudah mengenali suatu objek.
3. Resik: usaha membersihkan tempat kerja dan selalu mempertahankan kebersihan.
4. Rawat: usaha yang selalu memelihara keadaan ringkas, resik dan rapih.
5. Rajin: usaha yang menjadikan ringkas, rapi dan rajin menjadi kebiasaan.

Selain kebijakan kualitas, PT Krama Yudha Ratu Motor dalam aktifitasnya juga menerapkan kebijakan lingkungan, keselamatan dan kesehatan kerja. Hal tersebut diupayakan agar meraih keuntungan yang dapat bertahan dalam jangka panjang. Adapun tugas utama untuk mencapai tujuan tersebut dengan mengembangkan, membuat dan memasarkan produk dan jasa yang dapat memuaskan kebutuhan pelanggan secara aman dalam penggunaannya. Energi produk dan jasa tersebut juga digunakan secara efisien, tidak mencemari lingkungan dan dapat didaur ulang atau dibuang secara aman.

4.3 Visi dan Misi Perusahaan

Visi dari PT Krama Yudha Ratu Motor adalah:

1. Untuk pembangunan dan kesejahteraan masyarakat Indonesia.
2. Menjadi perusahaan yang mempunyai reputasi tinggi.
3. Menjadi perakitan mobil Mitsubishi yang terkenal di luar Jepang.
4. Dalam kurun waktu berikutnya sudah mampu merakit mobil.

Misi dari PT Krama Yudha Ratu Motor adalah:

1. Menjadi perusahaan yang paling menguntungkan dalam kalangan industri yang sama.
2. Menghasilkan keuntungan yang cukup memuaskan bagi *shareholders* dan kesejahteraan yang baik bagi seluruh karyawan.
3. Memperlakukan seluruh karyawan dengan cara adil dan terhormat.
4. Menjadi perusahaan yang kuat bertumbuh serta siap menghadapi untuk menghadapi persaingan regional maupun global.

Tugas dari PT Krama Yudha Ratu Motor adalah:

1. Tugas Jangka Pendek:
 - a) Meningkatkan volume produksi (ekspansi, investasi baru, dll).
 - b) Mengurangi biaya.
 - c) Menjaga kepatuhan.
2. Tugas Jangka Menengah-Panjang:
 - a) Memperbaiki dan meningkatkan kesadaran serta motivasi kerja.
 - b) Mempelajari dunia pabrik baru dan mengatur lini di pabrik berdasarkan pada model produksi masa depan dan volume produksi.

4.4 Manajemen Mutu

Jajaran Manajemen PT Krama Yudha Ratu Motor menetapkan dan memastikan penerapan sasaran perusahaan dan target departemen/bagian. Sasaran perusahaan ditetapkan untuk menjadi target yang akan dicapai oleh departemen dalam kurun waktu yang ditetapkan. Sasaran Mutu, sasaran lingkungan dan target bagian harus memenuhi sifat-sifat *SMART* :

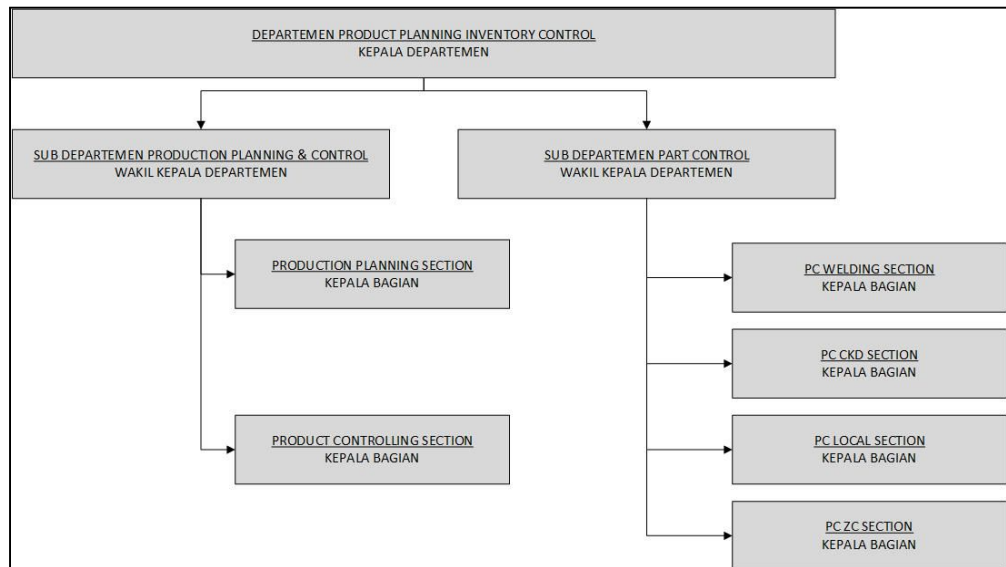
1. *Spesific* : Detail dan fokus
2. *Measurable* : Terukur
3. *Relevan & realistic* : Sesuai/ tidak mengada-ada
4. *Time frame* : Terjadwal dengan jelas

Dalam pelaksanaannya sasaran perusahaan di evaluasi setiap 1(satu) tahun sekali sedangkan target bagian dievaluasi setiap 3(tiga) bulan sekali.

4.5 Struktur Organisasi Perusahaan

Untuk menjalankan usahanya, setiap perusahaan memerlukan suatu struktur organisasi yang baik. Struktur organisasi dapat diartikan sebagai susunan dan hubungan antar bagian dan posisi dalam suatu perusahaan. Suatu struktur organisasi menggambarkan pembagian kerja, pelimpahan wewenang, kesatuan perintah dan tanggung jawab yang jelas.

Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan pencapaian dan target suatu perusahaan di dalam mencapai tujuannya. Struktur organisasi pada Gambar IV.2 dapat diartikan sebagai susunan dan hubungan antar bagian dan posisi dalam suatu perusahaan. Suatu struktur organisasi menggambarkan pembagian kerja, pelimpahan wewenang, kesatuan perintah dan tanggung jawab yang jelas.



1.14

1.15 **Gambar IV.3 Struktur Organisasi Divisi PPIC PT Krama Yudha Ratu Motor**

1.16 **Sumber: PT Krama Yudha Ratu Motor (2017)**

Berikut adalah uraian tugas kerja dan tanggung jawab kerja dari masing-masing jabatan yang tercantum pada struktur organisasi di atas:

1. Wakil Kepala Departemen.

Tugas dan tanggung jawab dari jabatan tersebut meliputi:

- a) Membantu/mewakili Kepala Departemen membuat rencana dan melaksanakan pengawasan serta pengendalian pada Bagian PPC, *Part Control & Information Technology* (IT).
- b) Membantu/mewakili Kepala Departemen memberdayakan dan memonitor kegiatan proses produksi dengan menggunakan tenaga kerja yang sesuai kondisi alat dan mesin agar mencapai produktivitas sesuai standar yang berlaku di dalam sebuah perusahaan.
- c) Membantu/mewakili Kepala Departemen mengevaluasi dan mengembangkan Sumber Daya Manusia (SDM) secara terpadu untuk mencapai sasaran yang telah ditetapkan.
- d) Melaksanakan tugas Kepala Bagian, bila Kepala Bagian mendapat tugas keluar perusahaan/berhalangan.
- e) Menjalankan, memelihara, meningkatkan sistem manajemen perusahaan.

f) Melakukan identifikasi, mengelola dan *monitoring* lingkungan ditempat kerja masing-masing.

2. Kepala Bagian 1

Tugas dan tanggung jawab dari jabatan tersebut meliputi:

- a) Menerima *order* produksi dari PT KTB (Krama Yudha Tiga Berlian).
- b) Membuat *Daily Production Schedule* terhadap 4 titik hitung yaitu: *Welding, Trimming On, Trimming Off & Delivery* dan mendistribusikan ke bagian terkait dan PT KTB.
- c) Membuat rencana produksi *spare parts* PT KTB.

3. Kepala Bagian 2

Tugas dan tanggung jawab dari jabatan tersebut meliputi:

- a) Mengontrol laju proses produksi dan sebagai penghubung ke bagian terkait jika ada hambatan dan perubahan-perubahan.
- b) Membuat laporan bulanan dan tahunan seluruh kegiatan produksi.
- c) Membuat laporan hasil produksi *spare parts* ke Departemen Keuangan untuk selanjutnya dipakai sebagai alat penagihan.
- d) Memberdayakan dan *monitoring* kegiatan proses produksi dan menggunakan tenaga kerja yang cukup sesuai kondisi alat dan mesin agar mencapai produktivitas sesuai standar.
- e) Memelihara lingkungan, alat-alat, mesin-mesin dengan melaksanakan program K3, 5R dan GKM.

4. Kepala Seksi

Tugas dan tanggung jawab dari jabatan tersebut meliputi:

- a) Membuat *hourly schedule* terhadap empat titik hitung antara lain: *Welding, Trimming On, Trimming Off & Delivery* dan mendistribusikannya.
- b) Mencatat hambatan-hambatan proses produksi dan melaporkan kepada atasan.
- c) Membuat *schedule spare parts* PT KTB.
- d) Membuat laporan tahunan seluruh kegiatan hasil produksi.

- e) Membuat laporan hasil produksi *spare parts* bulanan ke Departemen Keuangan.
 - f) Mengevaluasi, mengarahkan dan membina karyawan sehingga menghasilkan karyawan yang bertanggung jawab, disiplin dan berprestasi guna tercapainya sasaran mutu.
5. Kepala Urusan *Regular/Truck*
- Tugas dan tanggung jawab dari jabatan tersebut meliputi:
- a) Menerima informasi dari bagian *Part Control* mengenai kondisi kuantitas *part* kendaraan jenis *regular/truck* melalui catatan di papan tulis dan pemberitahuan lisan.
 - b) Merekap informasi hasil produksi kendaraan jenis *regular/truck* dari laporan harian *Foreman* Bagian *Welding, Painting, Trimming* dan *Quality Control*.
 - c) Mengecek jumlah *cabin & body* kendaraan jenis *regular/truck* apabila hasil produksi tidak mencapai target yang ditetapkan.
 - d) Memberi informasi kepada *Subsection Production Planning* untuk segera melakukan penjadwalan ulang jika terdapat masalah produksi dan kekurangan jumlah *part*.
 - e) Memberi informasi mengenai jika ada perubahan jadwal produksi kepada semua pihak/bagian yang terkait.
 - f) Mengendalikan, mengontrol dan memastikan produksi sesuai dengan jadwal yang sudah ditetapkan.
 - g) Membuat laporan hasil produksi kendaraan jenis *regular/truck*.
6. Kepala Urusan *ZC (Outlander Sports)*
- Tugas dan tanggung jawab dari jabatan tersebut meliputi:
- a) Menerima informasi dari *Part Control* mengenai kondisi kuantitas *part* kendaraan jenis *ZC* melalui catatan di papan tulis dan pemberitahuan lisan.
 - b) Merekap informasi hasil produksi kendaraan jenis *ZC* dari laporan harian *Foreman* Bagian *Welding, Painting, Trimming* dan *Quality Control*.

- c) Mengecek jumlah *cabin & body* kendaraan jenis *ZC* apabila hasil produksi tidak mencapai target yang ditetapkan.
- d) Memberi informasi kepada *Subsection Production Planning* untuk segera melakukan penjadwalan ulang jika terdapat masalah produksi dan kekurangan jumlah *part*.
- e) Memberi informasi mengenai jika ada perubahan jadwal produksi kepada semua pihak/bagian yang terkait.
- f) Mengendalikan, mengontrol dan memastikan produksi sesuai dengan jadwal yang sudah ditetapkan.
- g) Membuat laporan hasil produksi kendaraan jenis *ZC*.

4.7 Analisis Dokumen Sistem Penjadwalan Produksi

Divisi PPIC mempunyai tanggung jawab penuh terhadap penjadwalan produksi. Penjadwalan produksi dilaksanakan setiap bulan untuk memenuhi kebutuhan pemesanan yang ada. Dalam proses penjadwalan produksi ini dibutuhkan beberapa dokumen agar proses produksi dapat berjalan sesuai dengan target yang ada. Dokumen-dokumen yang terlibat dalam sistem penjadwalan produksi *Colt T120ss* adalah sebagai berikut:

4.7.1 Dokumen Masuk

1. Form CKD Simulation

CKD Simulation (MFTBC Product)																																	
Model	End Sep'17	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	End Oct'17
JTB	Qty	60																															
	SlNo	6360	6361	6362	6363	6364	6365	6366	6367	6368	6369	6370	6371	6372	6373	6374	6375	6376	6377	6378	6379	6380	6381	6382	6383	6384	6385	6386	6387	6388	6389	6390	
	Lot No	105	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106	106
JTB	Qty	107																															
	SlNo	190																															
	Lot No	107																															
JTB	Qty	190																															
	SlNo	6140	6141	6142	6143	6144	6145	6146	6147	6148	6149	6150	6151	6152	6153	6154	6155	6156	6157	6158	6159	6160	6161	6162	6163	6164	6165	6166	6167	6168	6169	6170	
	Lot No	1024	1024	1024	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	1025	
JTB	Qty	190																															
	SlNo	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	
	Lot No	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	
JTB	Qty	39																															
	SlNo	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	
	Lot No	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	188	

Gambar IV.4 Form CKD Simulation
(Sumber: PT Krama Yudha Ratu Motor, 2017)

Deskripsi : Divisi PPIC mengidentifikasi kebutuhan pesanan *Colt T120ss* dengan cara rapat dengan *customer* (PT Krama Yudha Tiga Berlian) dan mendapatkan kesepakatan dengan mengisi pada kolom *Plan*.

Struktur Dokumen : Rangkap 2 (dua)

Rangkap 1 untuk arsip pada KTB

Rangkap 2 untuk diberikan kepada bagian PPIC

Asal : PT Krama Yudha Tiga Berlian (*Customer*)

Tujuan : Kepala Divisi PPIC

Frekuensi : 1 kali / tahun, namun revisi diperkirakan dapat 2 kali / bulan

Kolom-kolom :

- Model* : Berisi kode produksi mobil.
- Quantity* : Berisi banyaknya produksi yang harus dilakukan pada hari itu.
- S/No.* : Berisi *Serial Number* dari mobil.
- C/C* : Berisi jumlah akhir produksi pada akhir setiap minggu
- Date* : Berisi tanggal dokumen tersebut dibuat
- Dokumen : Berisi nomor dokumen tersebut

2. Form Man Hour

PT KRAMA YUDHA RATA MOTOR DPL/PPIC/PPC		MAN HOUR / TYPE / MONTH : Januari ~ Desember 2017												008504000 - F001 - PPC-7.1-1 Tanggal Berlak : 01 - JANUARI - 2017 Revisi : 01 - JUNI - 2017																
Section	Model	January		February		March		April		May		June		July		August		September		October		November		December		W/Average	T			
		Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW	Prod	DW					
Welding Plan	Colt	5,0	2,3	4,3	5,6	2,3	4,1	5,6	2,3	4,1	4,9	2,1	4,3	5,0	2,3	4,1	5,6	2,3	4,3	6,0	2,7	4,0	7,4	2,9	2,6			4,1		
	CRSL	10,7	2,0	2,0	10,7	2,0	2,0	10,7	2,0	2,0	9,8	2,7	2,8	10,2	2,0	2,8	10,2	2,0	2,0	10,2	2,0	2,8	10,2	2,0	2,8			2,8		
	RRSL	10,7	2,0	2,0	10,7	2,0	2,0	10,7	2,0	2,0	9,8	2,7	2,8	10,2	2,0	2,8	10,2	2,0	2,0	10,2	2,0	2,8	10,2	2,0	2,8			2,8		
	T0 side	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0	1,0	0,7	1,0			1,0		
	T0 side	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1	1,1	0,8	1,1			1,1		
Metal Finish Plan	Colt	5,9	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3	6,0	1,8	2,3			2,3		
	CRSL	1,00	0,1	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11	12,00	0,10	11			11,9
	RRSL	10,7	4	3,3	10,7	4	3,3	10,7	4	3,3	10,7	4	3,3	10,2	3	3,3	10,2	3	3,3	10,2	3	3,3	10,2	3	3,3			3,3		
	T0 side	1,0	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4	0,4			0,4		
	T0 side	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1	1,1	0,4	1,1			1,1		
IC	1,00	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05	0	0,00	0,05			0,0		

Gambar IV.5 Form Man Hour

(Sumber: PT Krama Yudha Ratu Motor, 2017)

Deskripsi : Sub divisi *Product Planning* mengidentifikasi waktu yang dapat diperoleh dalam menyelesaikan 1 produk mobil tiap pekerjaannya.

Struktur Dokumen : Tidak rangkap

Asal : Divisi PPIC
 Tujuan : Bagian Produksi
 Frekuensi : 1 kali / bulan, namun diperkirakan revisi 6 kali / bulan
 Kolom-kolom :

- a. *Model* : Berisi kode produksi mobil.
- b. *Section* : Berisi pada tahap mana *Man Hour* tersebut
- c. *Total* : Berisi Jumlah produksi mobil bulan tersebut
- d. *Prod/D* : Berisi jumlah produksi harian
- e. *Date* : Berisi tanggal dokumen tersebut dibuat
- f. *Dokumen* : Berisi nomor dokumen tersebut

2. *Form Daily Production Schedule*

PT. KRAMA YUDHA RATU MOTOR Departemen PPIC / PPC		RENCANA PRODUKSI HARIAN										Dokumen : FRM - PPC - 7.3 - 010				
TITIK HITUNG : WELDING CUM												Tanggal Berlaku : 27/01/2009				
												Revisi : 00				
												TANGGAL : 4-Jul-17				
JAM KE	6															
07.10 - 08.10	6															
08.10 - 09.10	7															
09.10 - 10.10	7															
10.10 - 11.10	7															
11.10 - 12.10	3															
12.10 - 13.10	7															
13.10 - 14.10	7															
14.10 - 15.10	7															
15.10 - 16.10	7															
16.10 - 17.10																
17.10 - 18.10																
18.10 - 19.10																
19.10 - 20.10																
TOTAL	51															
TYPE	CAM PA															

Gambar IV.7 *Form* Rencana Produksi Harian

(Sumber: PT Krama Yudha Ratu Motor, 2017)

Deskripsi : Divisi PPIC menghitung dan melaporkan produksi harian yang harus dilakukan pada setiap jamnya

Struktur Dokumen : Tidak rangkap

Asal : Divisi *Product Control*

Tujuan : Kepala Divisi PPIC dan Bagian produksi

Frekuensi : 1 kali / hari

Kolom-kolom :

- a. *Total* : Berisi *total* produksi harian.
- b. *Type* : Berisi tipe mobil yang akan di produksi
- c. *Date* : Berisi tanggal dokumen tersebut dibuat
- d. *Dokumen* : Berisi nomor dokumen tersebut
- e. *Jam Ke* : Berisi jumlah produksi pada jam tertentu

4.8 Analisis Proses (*Flowmap* Sistem yang Sedang Berjalan)

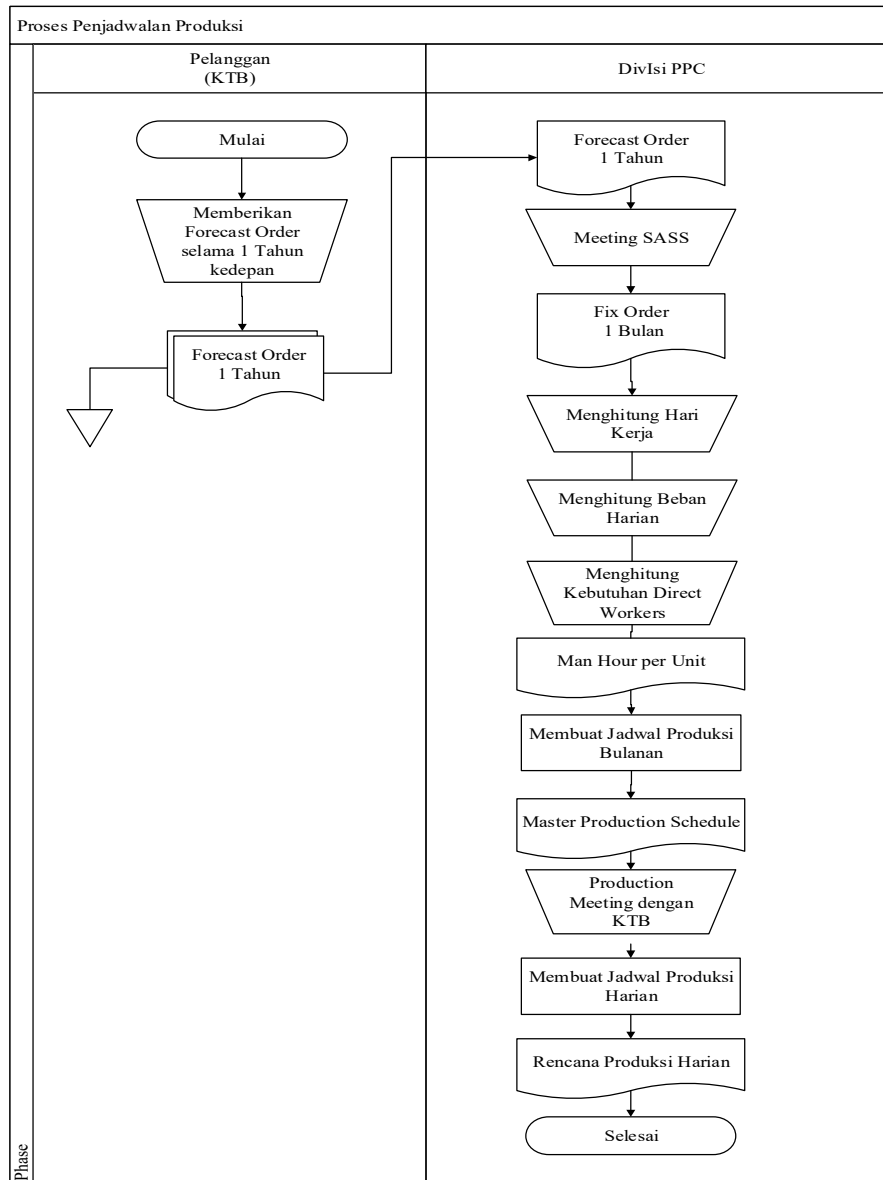
Bagan alir (*flowmap*) menunjukkan arus dari pekerjaan secara keseluruhan dari sistem. Bagan alir ini digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. *Flowmap* ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Adapun proses-proses tersebut dapat dijelaskan sebagai berikut:

4.9 *Flowmap* Proses Penjadwalan Produksi

Beberapa tahapan untuk dapat melaksanakan penjadwalan produksi. Adapun prosesnya sebagai berikut :

1. Krama Yudha Tiga Berlian (KTB) memberikan *forecast order* selama 1 tahun produksi yang akan dilakukan oleh Krama Yudha Ratu Motor.
2. Divisi PPIC menerima dokumen tersebut, lalu kepala divisi PPIC, serta wakil divisi PPC mengundang KTB untuk melakukan *meeting* SASS.
3. Kepala bagian *Product Planning* menghitung hari kerja bulan tersebut dari jumlah hari kerja pada bulan tersebut
4. Bagian *Product Planning* menghitung beban harian selama proses produksi menggunakan kalkulator
5. Menghitung beban harian *direct workers* dengan kalkulator
6. *Foreman Product Controlling* akan memasukkan data-data yang telah diperoleh sebelumnya untuk dimasukkan kedalam data *Master Production Schedule*.
7. Ketika MPS telah dibuat dan disetujui oleh kepala divisi PPIC, diadakan kembali *meeting* bersama KTB untuk mendapatkan kesepakatan jadwal produksi.
8. Jika sudah diperoleh kesepakatan Antara kedua belah pihak, maka *product planning Section* membuat jadwal produksi harian yang pada akhirnya akan menjadi form *Daily Production Schedule (DPS)*. Jadwal tersebut dibuat menggunakan *Microsoft Excel*.

Flowmap pada gambar IV.8 akan menjelaskan tentang proses penjadwalan produksi tersebut:



Gambar IV.8 *Flowmap* Proses Penjadwalan Induk Produksi

(Sumber: Hasil Pengamatan, 2017)

4.10 Analisis Permasalahan

Permasalahan yang ada pada sistem penjadwalan produksi *Colt T120ss* yang sedang berjalan pada PT Krama Yudha Ratu Motor dapat dilihat pada Tabel IV.1 berikut:

Tabel IV.1 Analisis Permasalahan

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
<i>Performan</i> (Kinerja)	Untuk melakukan proses perhitungan penjadwalan produksi yang di lakukan oleh <i>fore-man Planning</i> yang masih dilakukan menggunakan kertas dan masih <i>manual</i> menggunakan kalkulator.	Belum adanya sistem yang terkomputerisasi yang dapat membantu proses penjadwalan produksi	Proses penjad-walan produksi memakan waktu yang lama	Merancang dan membangun su-atu sistem yang membantu pro-ses seleksi penjadwalan produksi
<i>Informatio</i> <i>n</i> (Informasi)	Data penjadwalan produksi yang kurang terorganisir karena masih <i>print out</i> dan disimpan di dalam arsip	Masih menggunakan <i>Microsoft Excel</i>	Inkonsistensi da-ta revisi penjad-walan produksi, data dapat hilang	Membuat proses <i>inputan</i> data penjadwalan produksi menjadi terkompu-

	<i>file</i>			terisasi
<i>Economics</i>	-	-	-	-
<i>Control</i> (Pengendalian)	Sering terjadinya kesalahan dalam perhitungan proses penjadwalan produksi.	<i>Foreman</i> <i>PPC</i> masih kurang teliti dalam menghitung proses penjadwalan produksi.	Revisi yang harus dilakukan dalam menghitung penjadwalan produksi dilakukan berkali-kali	Menerapkan metode MPS (<i>Master Production Schedule</i>) yang sudah terkomputerisasi untuk mengurangi kesalahan dalam menghitung penjadwalan produksi.
<i>Efficiency</i>	-	-	-	-
<i>Service</i> (Layanan)	Proses penjadwalan produksi hanya dapat dilakukan oleh <i>Foreman</i> <i>PPC</i>	Belum ada sistem yang terkomputerisasi untuk membantu karyawan dalam menghitung penjadwalan produksi	Tidak adanya pengganti apabila <i>Foreman</i> <i>PPC</i> tidak masuk, lalu adanya revisi penjadwalan, maka revisi tersebut hanya dapat dilakukan	Merancang dan membangun suatu sistem yang dapat membantu dalam proses penjadwalan produksi.

			apabila <i>foreman PPC</i> masuk ker-ja kembali	
--	--	--	----------------------------------------------------------	--

(sumber: Hasil Analisis, 2018)

Untuk analisis permasalahan pada aspek *Economics*, dan *Efficiency* tidak didapatkan cukup data penunjang, sehingga analisis permasalahan tersebut belum dapat dilakukan.

BAB V

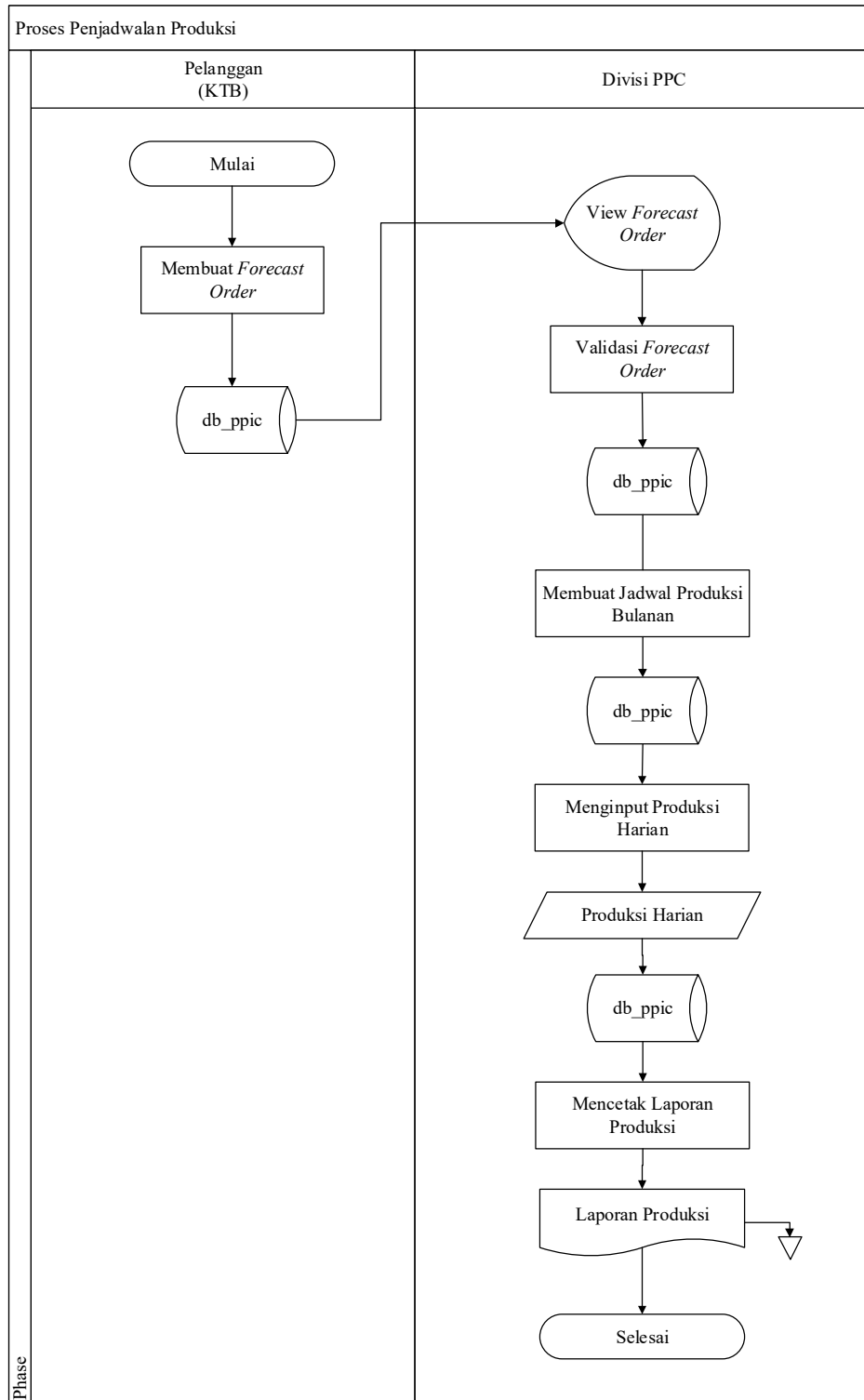
ANALISIS DAN PEMBAHASAN

5.1 Alur Proses Penjadwalan Produksi Usulan

Beberapa tahapan untuk dapat melaksanakan penjadwalan produksi. Adapun prosesnya sebagai berikut:

9. Krama Yudha Tiga Berlian (KTB) menginputkan *forecast order* kepada Krama Yudha Ratu Motor.
10. Divisi PPIC melihat data *forecast order* yang ada di *database*.
11. Divisi PPIC melakukan validasi *forecast order*, yang akhirnya menjadi *fix order* produksi.
12. Divisi PPIC membuat jadwal produksi bulanan yang telah di input oleh customer
13. Divisi PPIC menginput produksi harian yang sedang berlangsung.
14. Divisi PPIC mencetak laporan produksi dan akan menghasilkan dokumen laporan produksi.

Flowmap pada gambar IV.1 akan menjelaskan tentang proses penjadwalan produksi tersebut:



Gambar V.1 Flowmap Usulan Proses Penjadwalan Produksi Usulan
 Sumber: Hasil Analisis(2018)

5.2 Analisis Kebutuhan Sistem

Kebutuhan-kebutuhan dari sistem informasi penjadwalan produksi yang diusulkan dapat dilihat pada Tabel V.1 berikut:

Tabel V.1 Analisis Kebutuhan Sistem

No	Masalah	Kebutuhan User	Solusi	Kebutuhan Sistem	
				Functional Requirement	Non Functional Requirement
1	Untuk melakukan proses perhitungan penjadwalan produksi yang dilakukan oleh <i>foreman planning</i> yang masih dilakukan menggunakan kertas dan masih <i>manual</i> dengan kalkulator		Membuat sistem penjadwalan produksi yang mengeluarkan <i>output</i> berupa laporan penjadwalan produksi selama sebulan yang sudah terkomputerisasi.	Sistem dapat merekap penjadwalan produksi serta dapat mengelola dan mencetak laporan produksi bulanan.	Laporan penjadwalan produksi dikelola oleh <i>foreman</i> PPC.
2	Data penjadwalan produksi yang kurang terorganisir karena masih <i>print out</i> dan disimpan di dalam arsip <i>file</i> .	Sistem yang dapat melakukan validasi terhadap <i>Forecast Order</i> yang hanya dilakukan oleh kepala bagian PPIC.	Membuat sistem informasi berbasis komputer menggunakan <i>database</i> sebagai media penyimpanan.	Mengelola dan menyimpan data <i>mas-ter barang</i> , dan data CKD <i>Simulation</i> .	Data barang dikelola oleh pegawai PPC Data CKD <i>Simulation</i> dikelola oleh <i>Customer</i> .

Sumber: Hasil Analisis (2018)

Tabel V.1 Analisis Kebutuhan Sistem (Lanjutan)

No	Masalah	Kebutuhan User	Solusi	Kebutuhan Sistem	
				Functional Requirement	Non Functional Requirement
3	Sering terjadinya kesalahan dalam perhitungan proses penjadwalan produksi		Membuat sistem yang dapat menghitung penjadwalan produksi menggunakan metode <i>Master Production Schedule</i> (MPS)	Menyimpan & mengelola data penjadwalan produksi	Data penjadwalan produksi dikelola oleh pegawai PPC
4	Penjadwalan produksi hanya dapat dihitung oleh <i>foreman</i> PPC.		Menyediakan hak akses untuk Pegawai PPC agar dapat memasukkan data penjadwalan produksi.	Menyimpan & mengelola data <i>login</i>	Pembagian hak akses hanya dapat dilakukan oleh <i>admin</i> .

Sumber: Hasil Analisis (2018)

5.2.1. Functional Requirement

Functional requirement dari sistem informasi penjadwalan produksi yang diusulkan antara lain:

1. Pegawai PPC mengelola data *master* barang.
2. Pegawai Krama Yudha Tiga Berlian mengelola data CKD *Simulation*.
3. Kepala Bagian PPIC melakukan validasi *forecast order*.
4. Pegawai PPC mengelola data penjadwalan produksi.
5. *Foreman* PPC merekap data produksi.
6. *Foreman* PPC mencetak laporan produksi bulanan.

7. *Foreman* PPC, pegawai PPC, Krama Yudha Tiga Berlian, dan Kepala Bagian PPIC melakukan *Login*.
8. *Admin* mengelola hak akses kepada *user* lain yang akan menggunakan sistem tersebut.

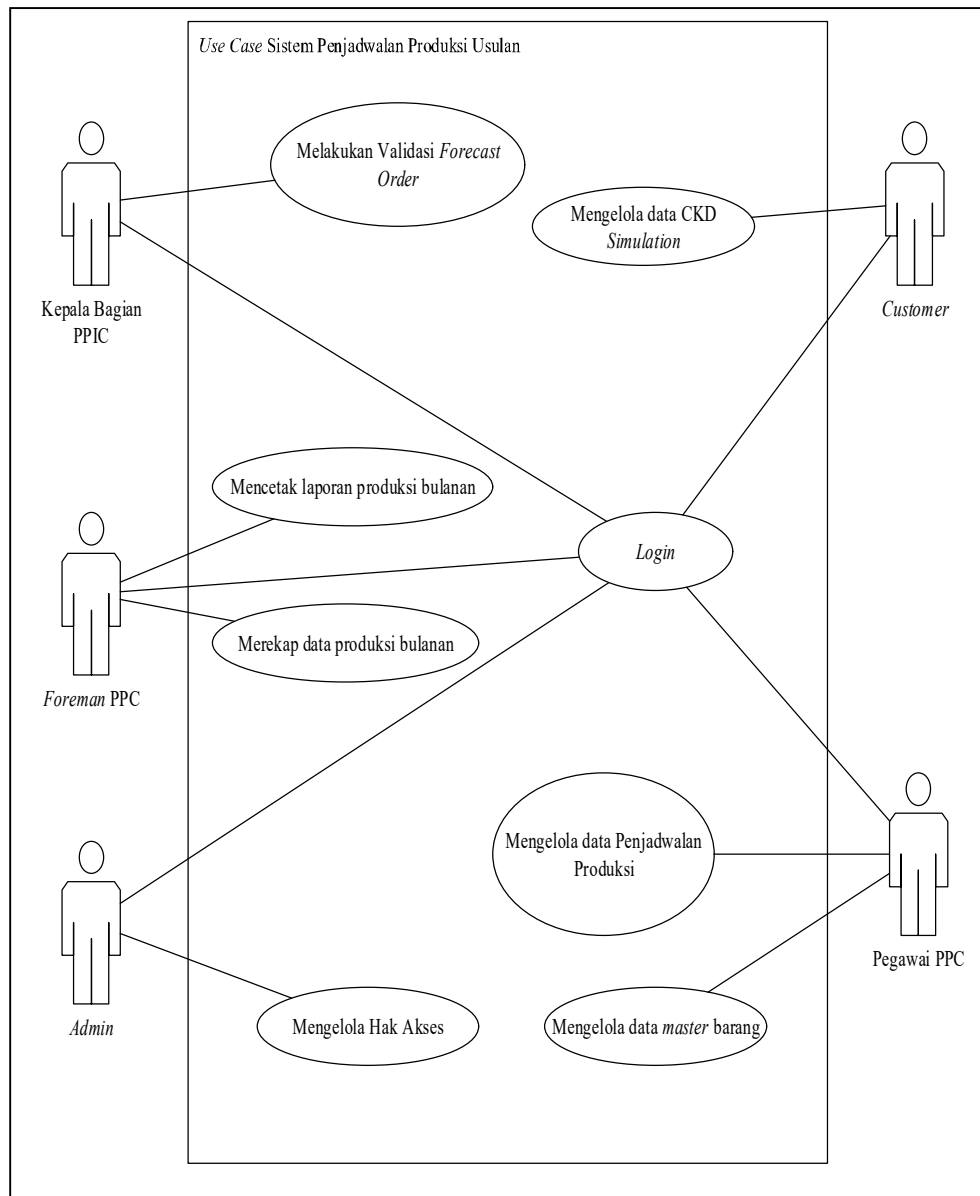
5.2.2. Non Functional Requirement

Non functional requirement dari sistem informasi penjadwalan produksi yang diusulkan antara lain:

1. Data penjadwalan produksi dikelola oleh pegawai PPC, proses pengelolaannya berupa menambah, menghapus, mengubah dan isi dari data tersebut adalah kode barang, nama barang, tanggal, *plan*.
2. Memberikan validasi terhadap *forecast order* yang telah diberikan oleh *customer* agar dapat menjadi *fix order* serta dijadwalkan setiap harinya.
3. Data laporan produksi bulanan dikelola oleh *foreman* PPC, proses pengelolaannya berupa menambah, mengubah, menghapus dan isi dari data tersebut adalah kode barang, nama barang, tanggal, *plan*, *actual*, *balance*
4. Pegawai PPC merekap hasil dari data produksi per hari.
5. Hasil dari rekap data, *Foreman* PPIC melakukan pencetakan laporan produksi bulanan.

5.3 Use Case Diagram

Use case diagram merupakan sebuah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dan apa yang diperbuat oleh sistem. Berikut merupakan *use case diagram* usulan penjadwalan produksi berdasarkan analisis kebutuhan sistem yang telah dibuat pada Tabel V.1:



Gambar V.2 Use Case diagram usulan Sistem Informasi Penjadwalan Produksi
(Sumber: Hasil analisis, 2018)

5.3.1 Use Case Description

Berikut merupakan *use case description* usulan sistem penjadwalan produksi dari *use case* usulan yang telah dibuat pada Gambar V.1:

Tabel V.2 *Use Case Description Login*

Nama <i>Use Case</i>	<i>Login</i>
Aktor	Kabag PPIC, Krama Yudha Tiga Berlian, <i>Admin</i> , Pegawai PPC
Deskripsi	<i>Use case</i> ini menggambarkan beberapa aktor dapat masuk kedalam sistem.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Aktor membuka <i>login form</i>. 2. Aktor memasukkan <i>username</i> dan <i>password</i> pada <i>login form</i>. 3. Sistem akan memvalidasi data yang dimasukkan apakah sudah sesuai dengan <i>database</i> atau tidak. 4. Jika sesuai, maka aktor akan masuk kedalam <i>homepage</i> sistem. 5. Jika salah, maka aktor akan mendapatkan <i>notification</i> pada <i>login form</i> yang menandakan bahwa data tidak sesuai.
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.3 *Use Case Description Mengelola CKD Simulation*

Nama <i>Use Case</i>	Mengelola CKD <i>Simulation</i>
Aktor	<i>Customer</i>
Deskripsi	<i>Use case</i> ini menggambarkan Krama Yudha Tiga Berlian memasukkan data CKD <i>Simulation</i> maupun mengubahnya.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. KTB memilih pilihan CKD <i>Simulation</i>. 2. Sistem akan menampilkan semua data CKD <i>Simulation</i> yang tersimpan pada <i>database</i>. 3. KTB dapat melihat data CKD yang tersimpan dan dapat juga menambahkan, mengubah, dan menghapus data CKD <i>Simulation</i>.
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.4 *Use Case Description* Mengelola Data Penjadwalan Produksi

Nama <i>Use Case</i>	Mengelola Data Penjadwalan Produksi
Aktor	Pegawai PPC
Deskripsi	<i>Use case</i> ini menggambarkan pegawai PPC mengelola data penjadwalan produksi.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pegawai PPC memilih menu CKD <i>Simulation</i>. 2. Sistem akan menampilkan data CKD <i>Simulation</i> yang tersimpan di dalam <i>database</i>. 3. Pegawai PPC memilih tombol jadwalkan produksi pada bulan yang dipilih. 4. Sistem akan mengambil semua informasi data penjadwalan produksi yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut kedalam tabel data penjadwalan produksi. 5. Pegawai PPC dapat melihat data penjadwalan yang tersimpan dan dapat menambahkan, dan mengubah, data penjadwalan produksi.
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.5 *Use Case Description* Melakukan Validasi *Forecast Order*

Nama <i>Use Case</i>	Melakukan Validasi <i>Forecast Order</i>
Aktor	Kepala Bagian PPIC (Kabag PPIC)
Deskripsi	<i>Use case</i> ini menggambarkan Kabag PPIC melakukan validasi <i>forecast order</i> yang telah di <i>input customer</i> .
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Kabag PPIC memilih menu CKD <i>Simulation</i>. 2. Sistem akan menampilkan data CKD <i>Simulation</i> yang tersimpan di dalam <i>database</i>. 3. Kabag PPIC akan memeriksa CKD <i>Simulation</i> yang telah di <i>input</i>, lalu Kabag PPIC menekan tombol validasi. 4. Sistem akan menyimpan di dalam <i>database</i> bahwa CKD <i>Simulation</i> telah divalidasi oleh Kepala Bagian PPIC

<i>Alternate Flow</i>	
-----------------------	--

Sumber: Hasil Analisis, 2018

Tabel V.6 *Use Case Description* Mengelola Data Master Barang

Nama <i>Use Case</i>	Mengelola Data <i>Master</i> Barang
Aktor	Pegawai PPC
Deskripsi	<i>Use case</i> ini menggambarkan pegawai PPIC mengelola data barang
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Pegawai PPIC memilih menu data <i>master</i>, lalu memilih pilihan data barang. 2. Sistem akan menampilkan semua data barang yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut pada tabel barang. 3. Pegawai PPIC dapat melihat data <i>master</i> barang yang tersimpan dan dapat juga menambahkan, mengubah, menghapus dan mencari data barang.
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.7 *Use Case Description* Merekap Laporan Produksi Bulanan

Nama <i>Use Case</i>	Merekap Laporan Produksi Bulanan
Aktor	<i>Foreman</i> PPC
Deskripsi	<i>Use Case</i> ini melakukan rekap data penjadwalan produksi bulanan
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Foreman</i> PPC memilih menu CKD <i>Simulation</i>. 2. Sistem akan menampilkan data CKD <i>Simulation</i> yang tersimpan di dalam <i>database</i>. 3. <i>Foreman</i> PPC memilih detail produksi pada bulan yang dipilih. 4. Sistem akan mengambil semua informasi data penjadwalan produksi yang tersimpan pada <i>database</i>. 5. <i>Foreman</i> PPC memilih tombol merekap data laporan produksi pada bulan yang sedang berjalan. 6. Sistem akan menampilkan data penjadwalan produksi yang diambil dari <i>database</i> pada bulan yang telah dipilih.

<i>Alternate Flow</i>	
-----------------------	--

Sumber: Hasil Analisis, 2018

Tabel V.8 *Use Case Description* Mencetak Laporan Produksi Bulanan

Nama <i>Use Case</i>	Mencetak Laporan Produksi Bulanan
Aktor	<i>Foreman PPC</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk membuat laporan hasil penjadwalan produksi
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Foreman PPC</i> memilih menu <i>CKD Simulation</i>. 2. Sistem akan menampilkan data <i>CKD Simulation</i> yang tersimpan di dalam database. 3. <i>Foreman PPC</i> memilih detail produksi pada bulan yang dipilih. 4. Sistem akan mengambil semua informasi data penjadwalan produksi yang tersimpan pada <i>database</i>. 5. <i>Foreman PPC</i> memilih tombol merekap data laporan produksi pada bulan yang sedang berjalan lalu memilih tombol cetak. 6. Sistem akan menampilkan data penjadwalan produksi yang diambil dari <i>database</i>, lalu akan menampilkan <i>print dialog</i>
<i>Alternate Flow</i>	

Sumber: Hasil Analisis, 2018

Tabel V.9 *Use Case Description* Mengelola Hak Akses

Nama <i>Use Case</i>	Mengelola Hak Akses
Aktor	<i>Admin</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk mengelola hak akses untuk karyawan.
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Admin</i> memilih menu data <i>master</i> lalu memilih data <i>login</i>. 2. Sistem akan mengambil semua informasi <i>login</i> yang tersimpan pada <i>database</i> dan menampilkan informasi tersebut kedalam tabel data login. 3. <i>Admin</i> dapat melihat detail <i>database</i> yang tersimpan dan

	dapat juga menambahkan, mengubah, dan menghapus data tabel login.
<i>Alternate Flow</i>	

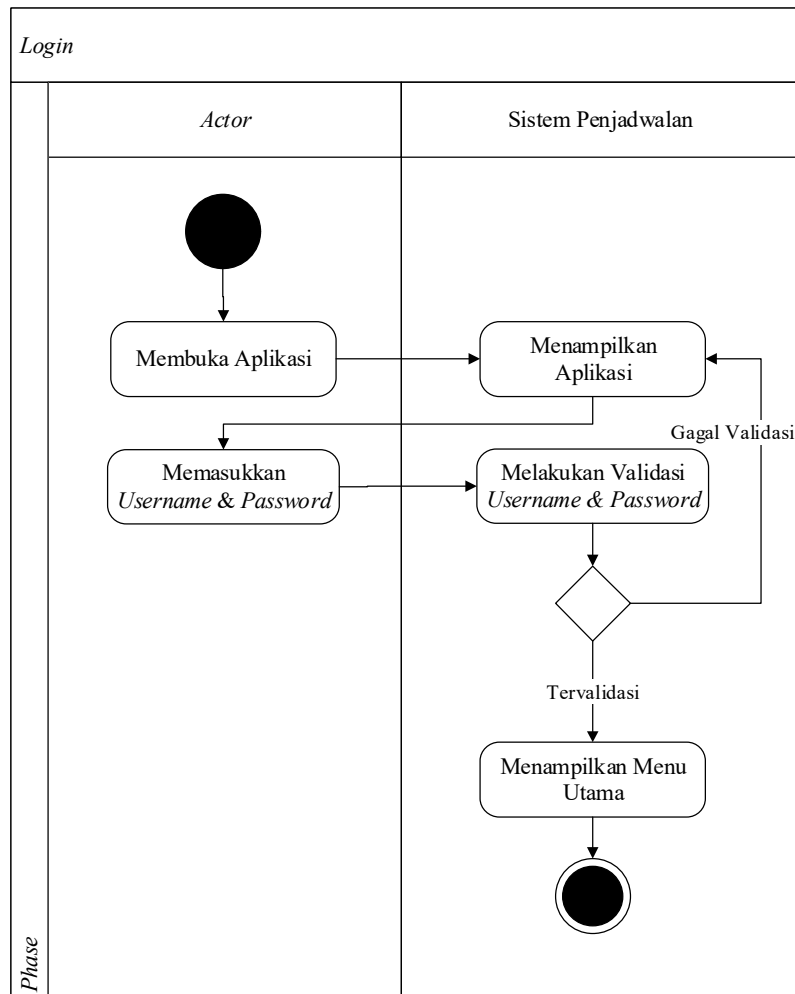
Sumber: Hasil Analisis, 2018

5.4 Activity Diagram

Activity diagram merupakan sebuah diagram yang menggambarkan urutan aktivitas proses bisnis pada suatu sistem. Berikut merupakan beberapa *activity diagram* usulan dari sistem informasi penjadwalan produksi:

1. *Activity Diagram Login*

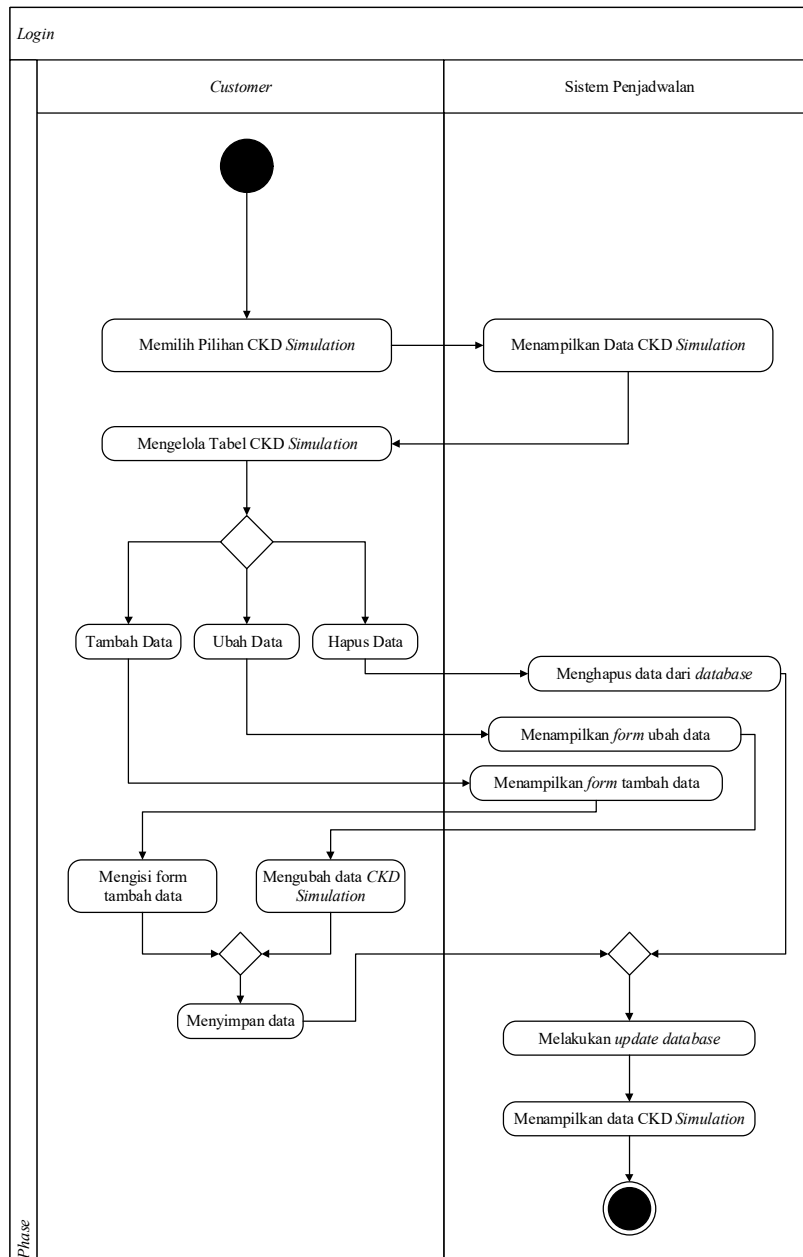
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu beberapa aktor seperti Kepala bagian PPIC, *foreman* PPC, *admin*, pegawai PPC, dan Krama Yudha Tiga Berlian untuk dapat mengakses sistem informasi penjadwalan produksi. Berikut merupakan *activity diagram login*:



Gambar V.3 Activity Diagram Login
 Sumber: Hasil Analisis(2018)

2. Activity Diagram Mengelola CKD Simulation

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh user, yaitu Krama Yudha Tiga Berlian untuk dapat mengelola data CKD Simulation yang berfungsi sebagai forecast order mulai dari tambah, ubah, maupun hapus.

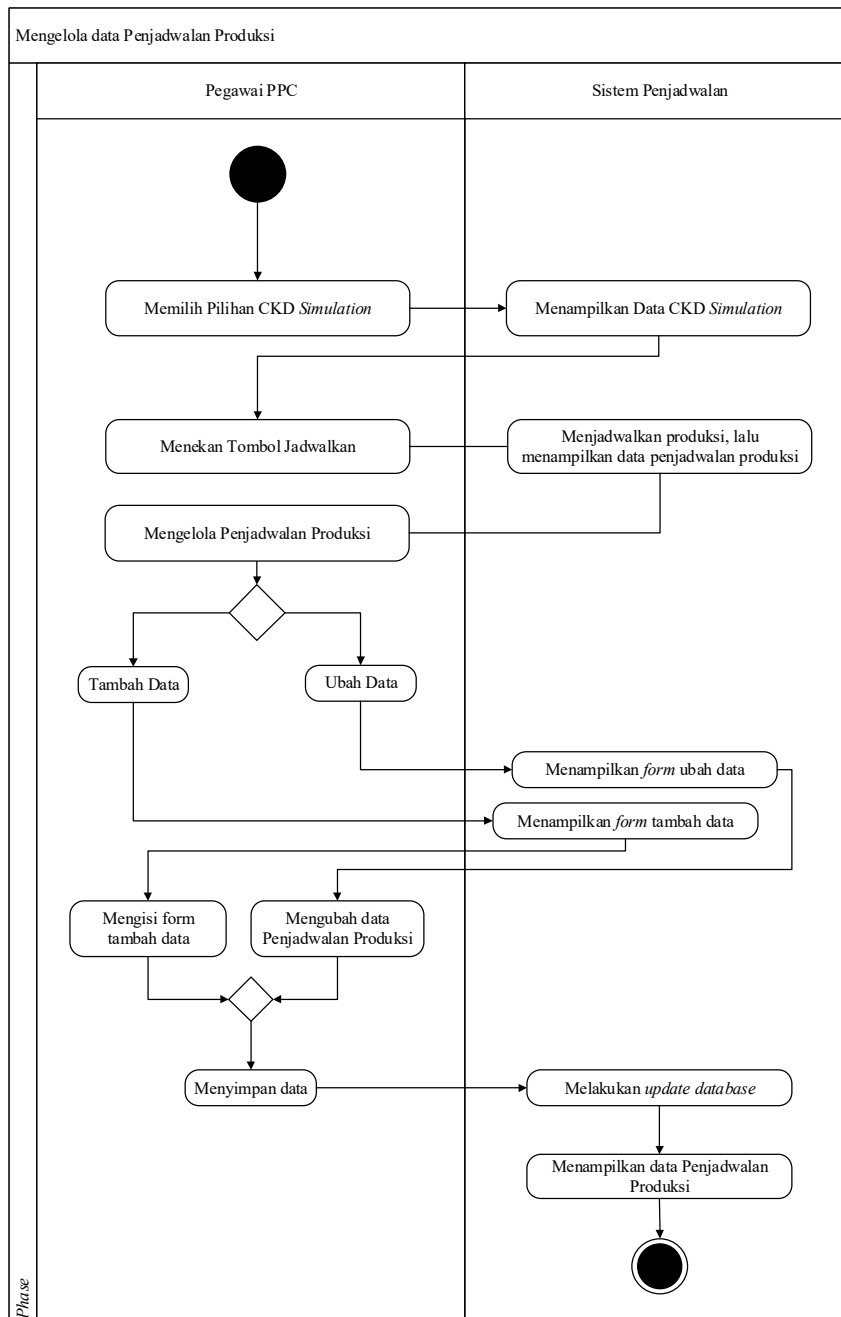


Gambar V.4 Activity Diagram Mengelola CKD Simulation

Sumber: Hasil Analisis(2018)

3. Activity Diagram Mengelola Data Penjadwalan Produksi

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh user, yaitu pegawai PPC untuk dapat mengelola data penjadwalan mulai dari tambah, ubah, maupun hapus.

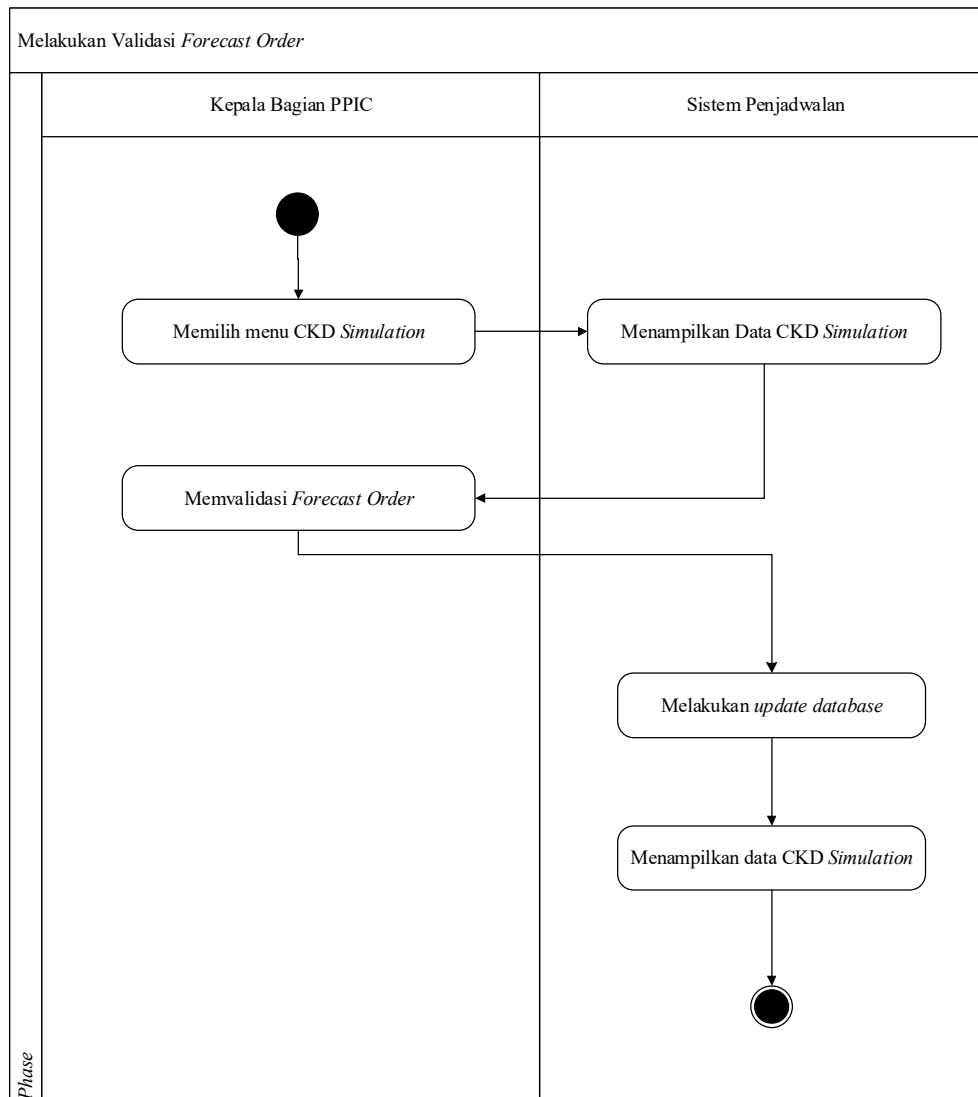


Gambar V.5 Activity Diagram Mengelola data Penjadwalan Produksi

Sumber: Hasil Analisis(2018)

4. Activity Diagram Melakukan Validasi Forecast Order

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh user, yaitu Kepala Bagian PPIC untuk dapat melakukan validasi forecast order yang telah dibuat oleh pegawai PPC.

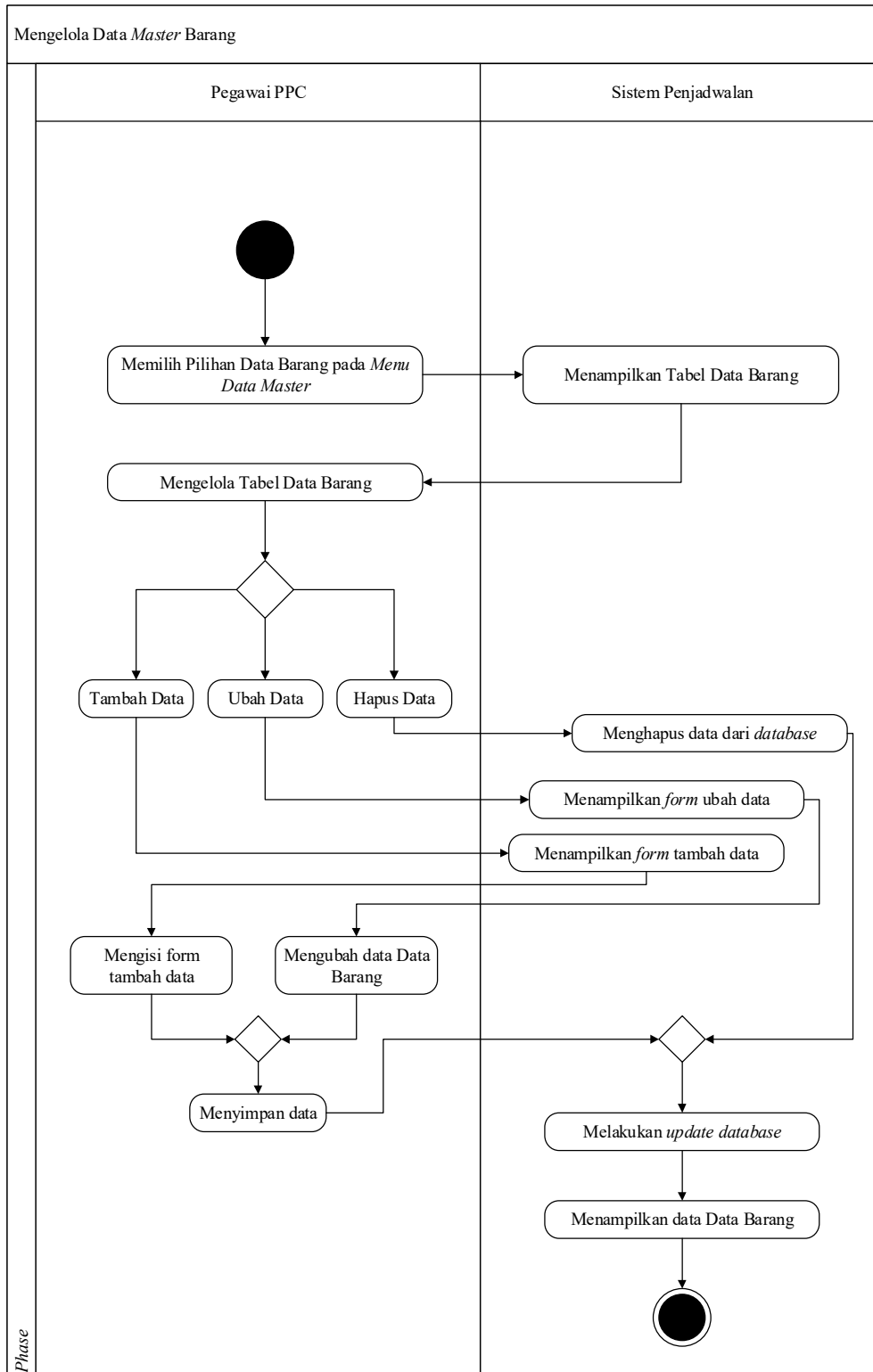


Gambar V.6 *Activity Diagram* Melakukan Validasi *Forecast Order*

Sumber: Hasil Analisis(2018)

5. *Activity Diagram* Mengelola Data *Master* Barang

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu pegawai PPC untuk dapat mengelola data barang mulai dari tambah, ubah, maupun hapus.

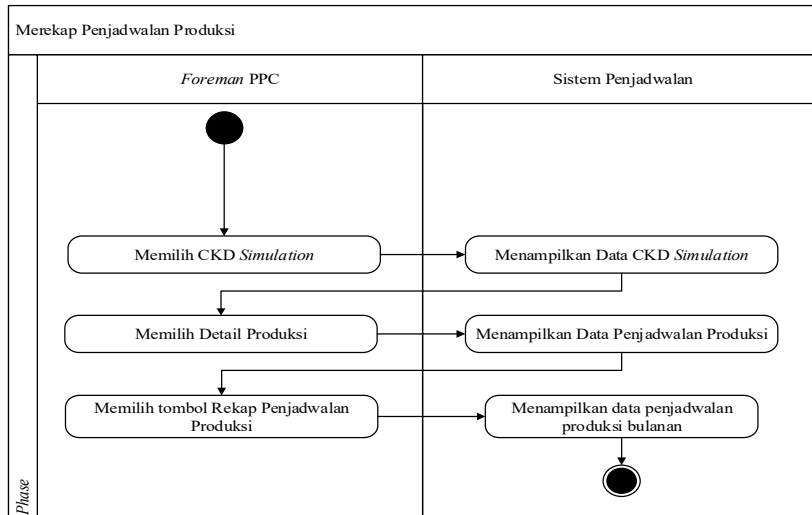


Gambar V.7 Activity Diagram Mengelola Data *Master* Barang

Sumber: Hasil Analisis(2018)

6. *Activity Diagram* Merekap Data Produksi Bulanan

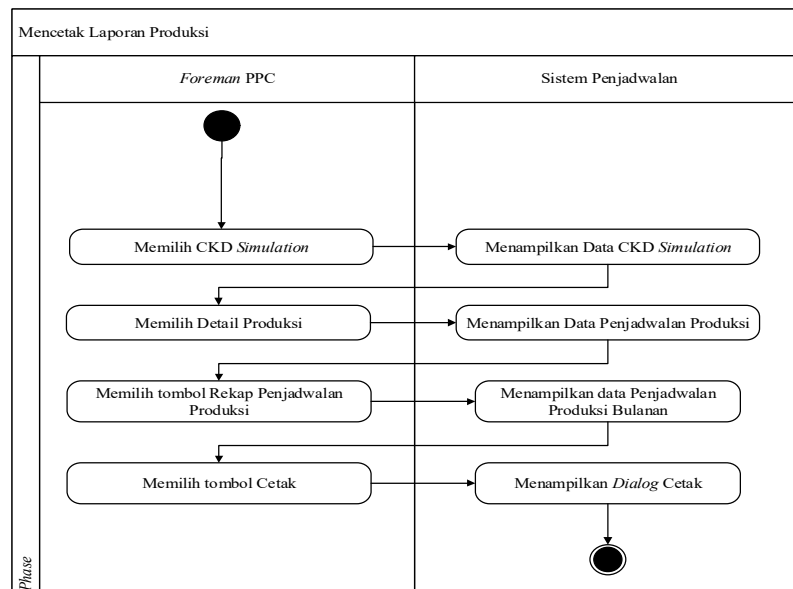
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu *foreman* PPC untuk dapat merekap data produksi yang telah selesai.



Gambar V.8 *Activity Diagram* Merekap Penjadwalan Produksi
Sumber: Hasil Analisis(2018)

7. *Activity Diagram* Mencetak Laporan Produksi Bulanan

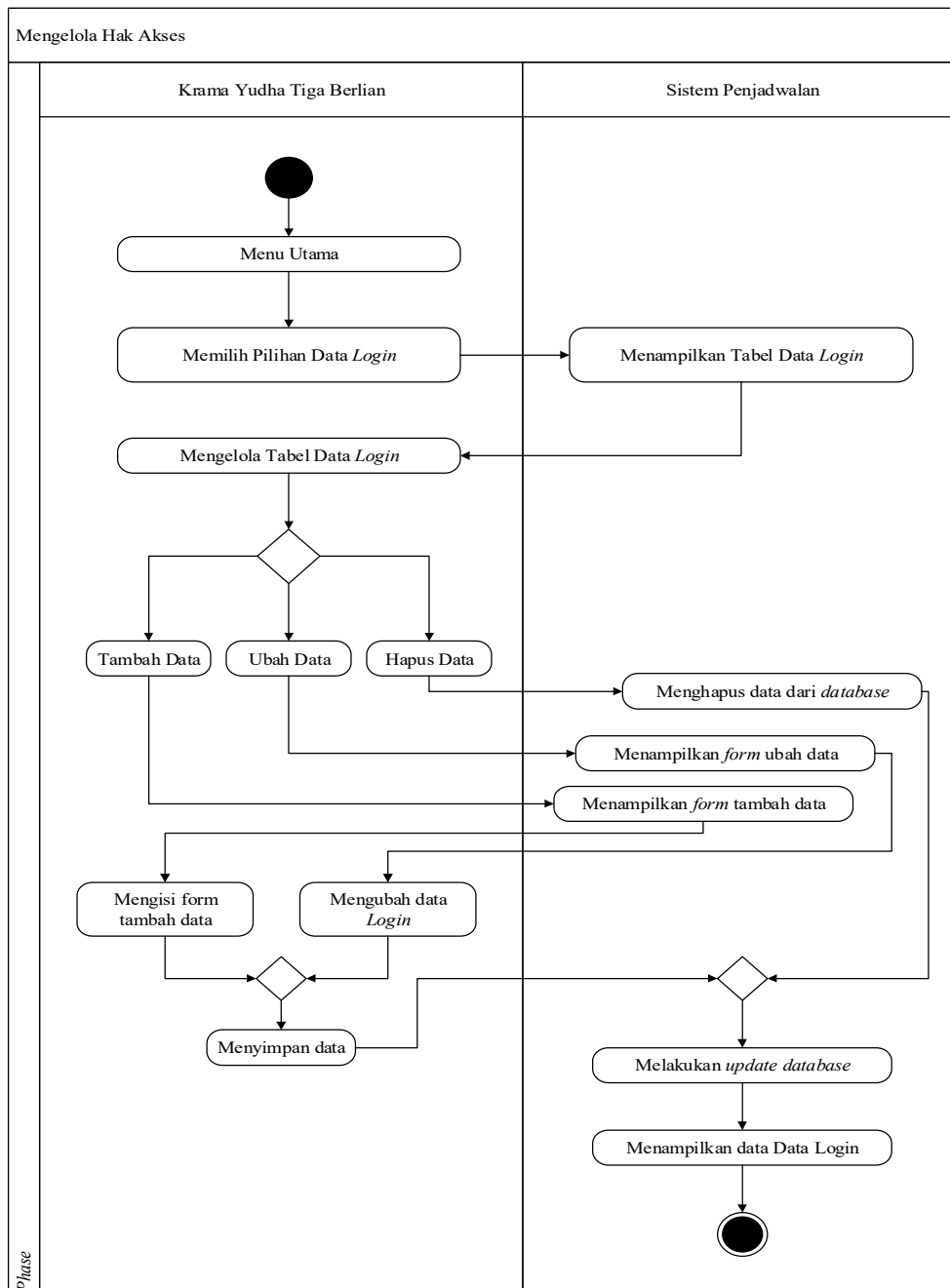
Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu *foreman* PPC untuk membuat laporan hasil produksi bulanan yang telah selesai.



Gambar V.9 *Activity Diagram* Mencetak Laporan Produksi
Sumber: Hasil Analisis(2018)

8. *Activity Diagram* Mengelola Hak Akses

Activity diagram ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu *Admin* untuk dapat mengelola data *login* serta hak akses kedalam sistem, mulai dari tambah, ubah, maupun hapus.



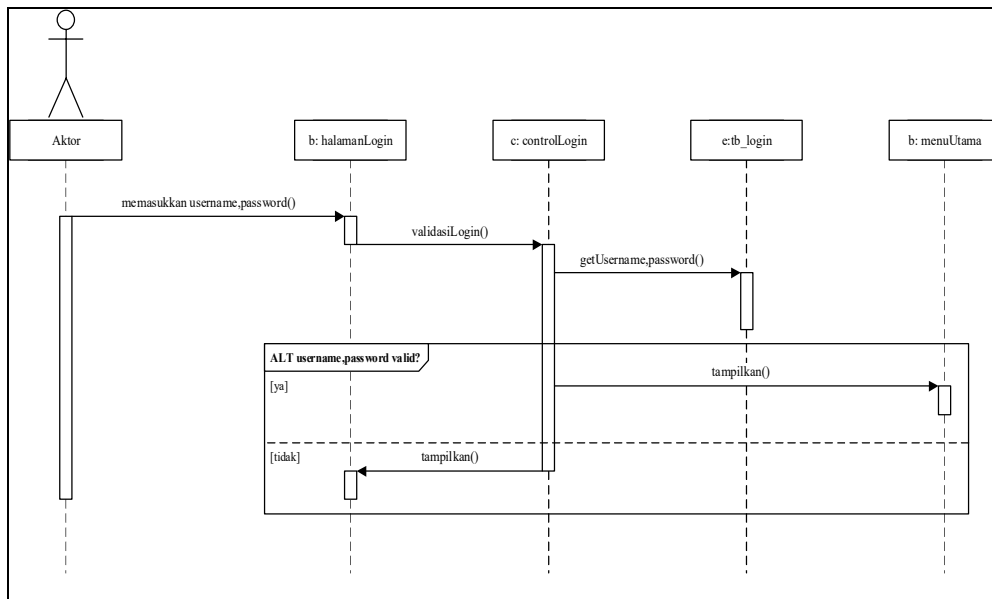
Gambar V.10 *Activity Diagram* Mengelola Hak Akses
 Sumber: Hasil Analisis(2018)

5.5 Sequence Diagram

Sequence diagram merupakan diagram untuk menggambarkan scenario ataupun rangkaian langkah yang dilakukan sebagai respon dari sebuah *event* dalam menghasilkan *output* tertentu. Berikut beberapa *sequence diagram* usulan dari sistem informasi penjadwalan produksi:

1. Sequence Diagram Login

Sequence Diagram Login merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses *login*. Di dalam sistem informasi penjadwalan produksi ini, *login* diperlukan agar seorang pengguna dapat mengakses sistem tersebut. Berikut merupakan *sequence diagram* dari *use case login*:

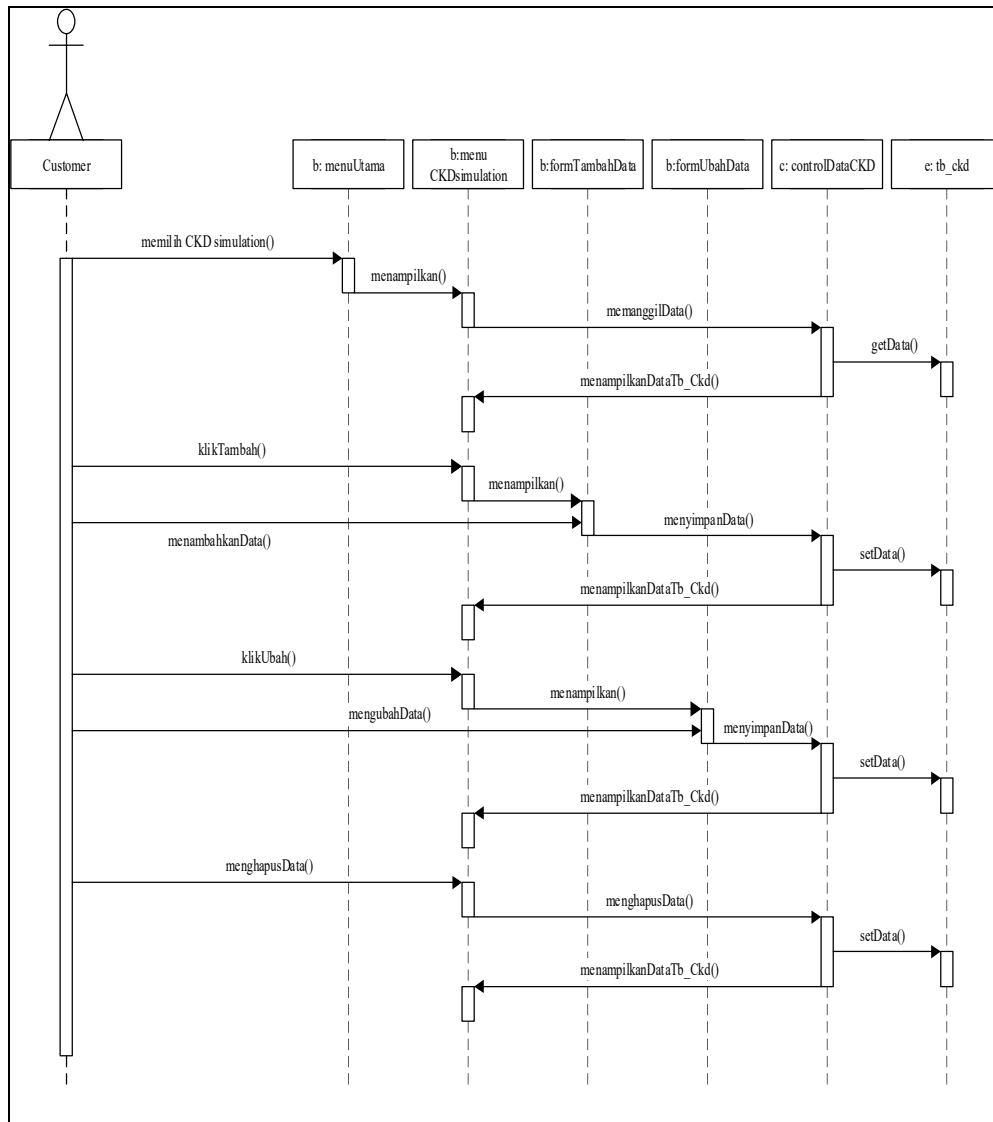


Gambar V.11 *Sequence Diagram Login*

Sumber: Hasil Analisis(2018)

2. Sequence Diagram Mengelola CKD Simulation

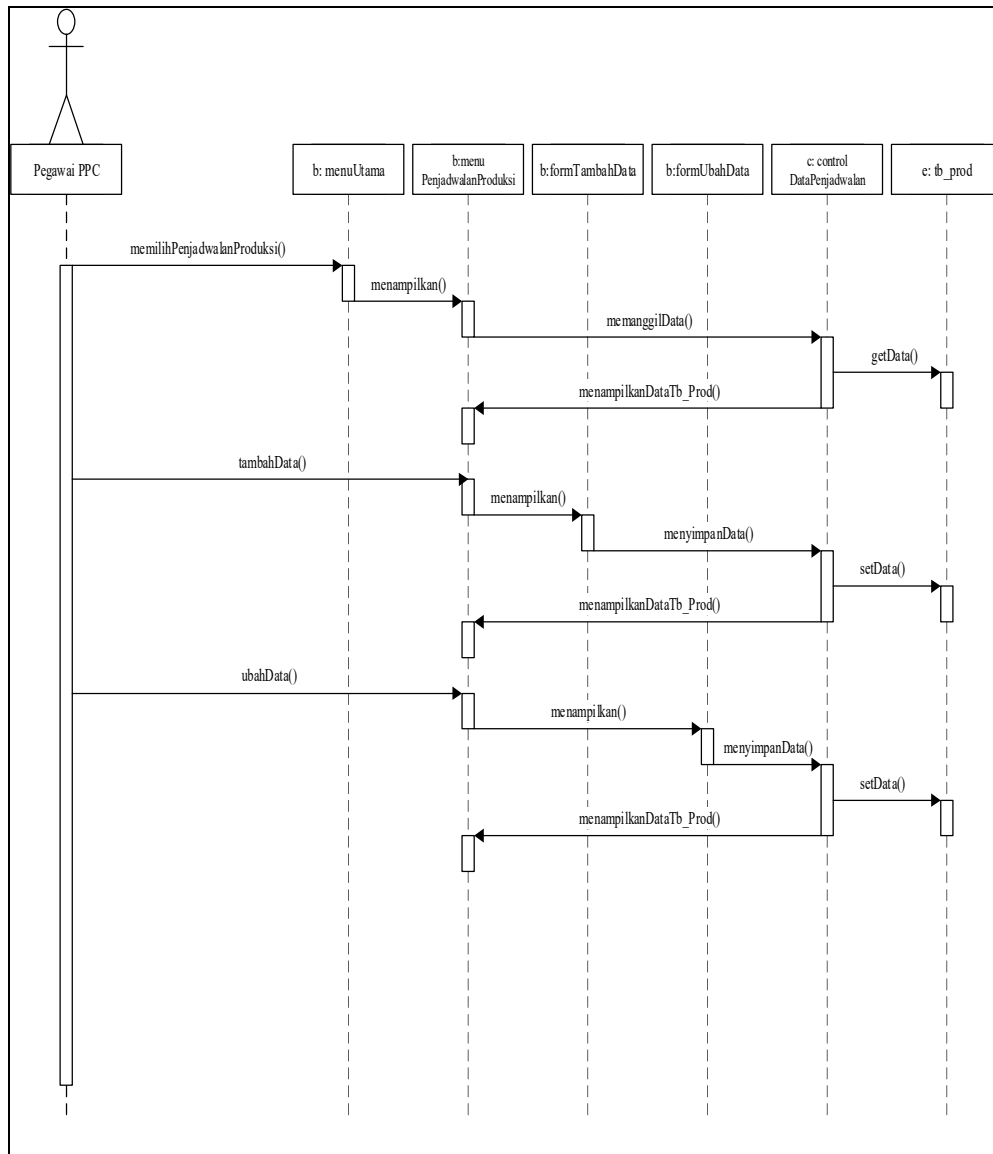
Sequence Diagram Mengelola CKD Simulation merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses mengelola CKD *Simulation*. Proses mengelola CKD *Simulation* ini dilakukan oleh *user*, yaitu Krama Yudha Tiga Berlian agar dapat menyimpan, menambah, mengubah, maupun menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola CKD *Simulation*:



Gambar V.12 *Sequence Diagram* Mengelola CKD Simulation
 Sumber: Hasil Analisis(2018)

3. *Sequence Diagram* Mengelola Data Penjadwalan Produksi

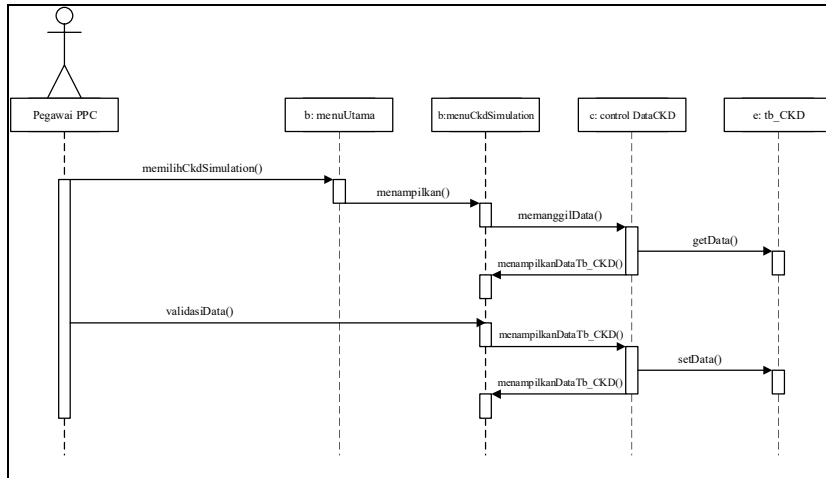
Sequence Diagram mengelola data penjadwalan produksi merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses mengelola data penjadwalan produksi. Proses mengelola data penjadwalan produksi ini dilakukan oleh pegawai PPC agar dapat menyimpan, menambah, mengubah, maupun menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data penjadwalan produksi:



Gambar V.13 *Sequence Diagram* Mengelola Data Penjadwalan Produksi
 Sumber: Hasil Analisis(2018)

4. *Sequence Diagram* Melakukan Validasi *Forecast Order*

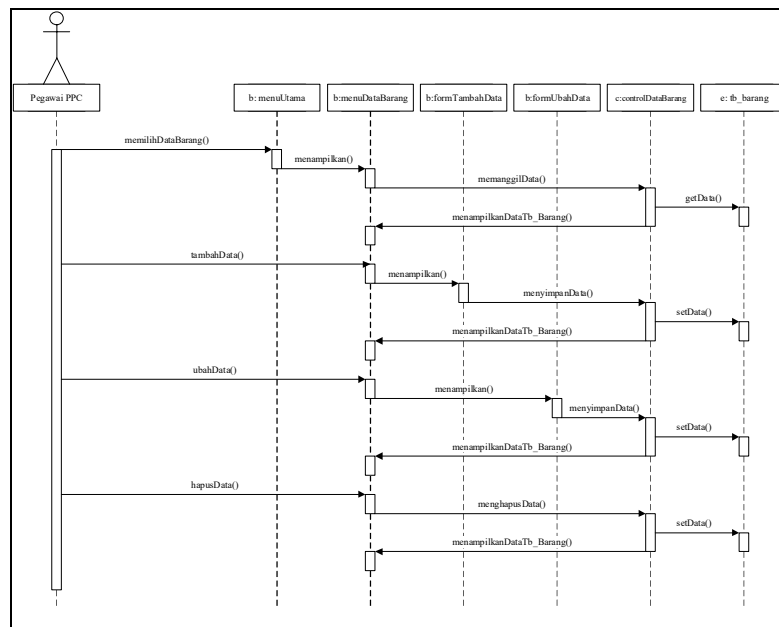
Sequence Diagram Melakukan Validasi *Forecast Order* merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses validasi. Proses melakukan validasi data *forecast order* ini dilakukan oleh *user*, yaitu kepala bagian PPIC agar dapat melakukan validasi *forecast order* yang telah dibuat oleh *customer*. Berikut merupakan *sequence diagram* dari use case melakukan validasi *forecast order*:



Gambar V.14 *Sequence Diagram* Melakukan Validasi Forecast Order
 Sumber: Hasil Analisis(2018)

5. *Sequence Diagram* Mengelola Data Master Barang

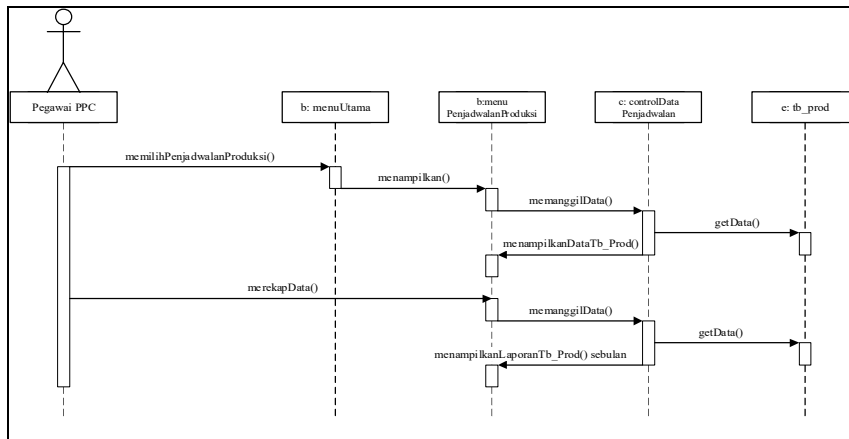
Sequence Diagram mengelola data *master* barang merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses mengelola data *master* barang. Proses mengelola data *master* barang ini dilakukan oleh *user*, yaitu pegawai PPC agar dapat menyimpan, menambah, mengubah, maupun menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola data *master* barang:



Gambar V.15 *Sequence Diagram* Mengelola Data Master Barang
 Sumber: Hasil Analisis(2018)

6. *Sequence Diagram* Merekap Laporan Produksi

Sequence Diagram merekup laporan produksi merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses tersebut. Proses ini dilakukan oleh user, yaitu *foreman* PPC agar dapat melakukan pengumpulan data hasil produksi pada bulan tersebut. Berikut merupakan *sequence diagram* dari *use case* merekup laporan produksi:

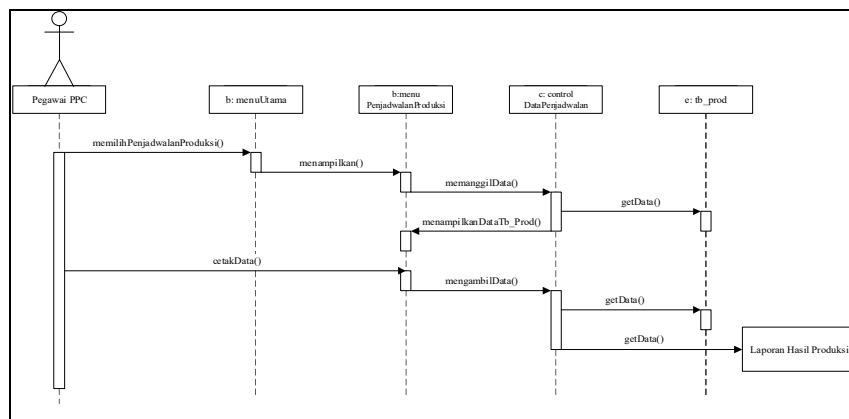


Gambar V.16 *Sequence Diagram* Merekup Laporan Produksi

Sumber: Hasil Analisis(2018)

7. *Sequence Diagram* Mencetak Laporan Produksi

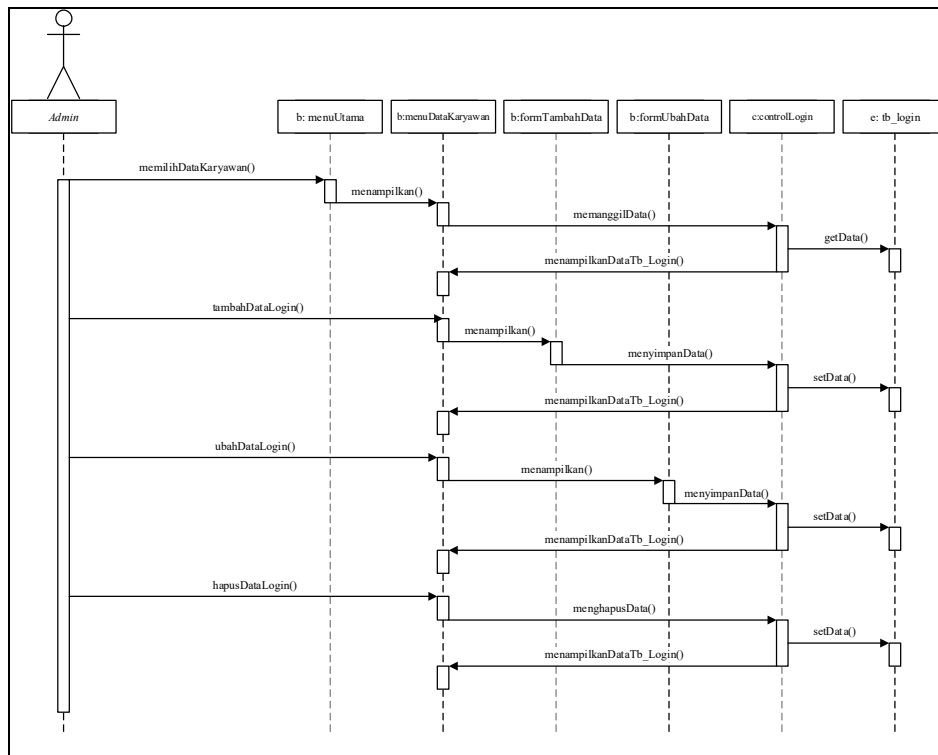
Sequence Diagram mencetak laporan produksi merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses tersebut. Proses ini dilakukan oleh *foreman* PPC agar dapat melakukan pencetakan laporan hasil produksi. Berikut merupakan *sequence diagram* dari *use case* mencetak laporan produksi:



Gambar V.17 *Sequence Diagram* Mencetak Laporan Produksi
 Sumber: Hasil Analisis(2018)

8. *Sequence Diagram* Mengelola Hak Akses

Sequence Diagram mengelola hak akses merupakan sebuah diagram *sequence* yang berfungsi untuk menjelaskan interaksi objek-objek dalam proses mengelola hak akses. Proses mengelola hak akses ini dilakukan oleh *user*, yaitu *admin* agar dapat menyimpan, menambah, mengubah, maupun menghapus data. Berikut merupakan *sequence diagram* dari *use case* mengelola hak akses:



Gambar V.18 *Sequence Diagram* Mengelola Hak Akses
 Sumber: Hasil Analisis(2018)

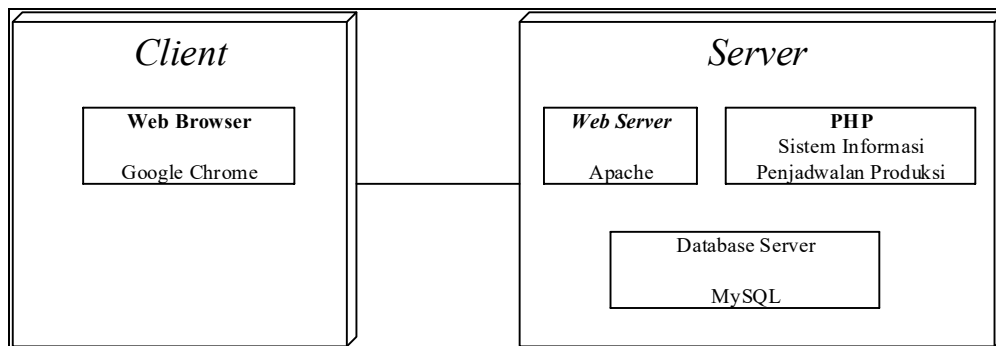
5.6 *Class Diagram*

Class Diagram pada usulan sistem informasi penjadwalan produksi ini digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat. Untuk membangun sistem, *class diagram* dapat memperlihatkan hubungan antar kelas dan penjelasan detail pada setiap kelas di

dalam model desain dari suatu sistem. Berikut merupakan *class diagram* usulan sistem informasi penjadwalan produksi:

5.7 *Deployment Diagram*

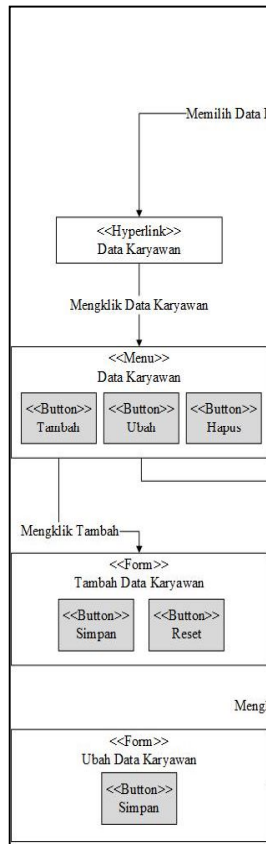
Deployment diagram pada usulan sistem informasi penjadwalan produksi ini digunakan untuk mewakili komponen-komponen perangkat lunak dan cara perangkat lunak tersebut ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Berikut ini merupakan *deployment diagram* sistem informasi penjadwalan produksi yang dapat dilihat pada gambar V.18:



Gambar V.20 *Deployment Diagram* Sistem Informasi Penjadwalan Produksi
Sumber: Hasil Analisis(2018)

5.8 *Windows Navigation Diagram (WND)*

Windows Navigation Diagram pada sistem usulan digunakan untuk menunjukkan bagaimana navigasi dari halaman-halaman yang terdapat di dalam aplikasi, berikut merupakan *Windows Navigation Diagram* sistem informasi penjadwalan produksi:



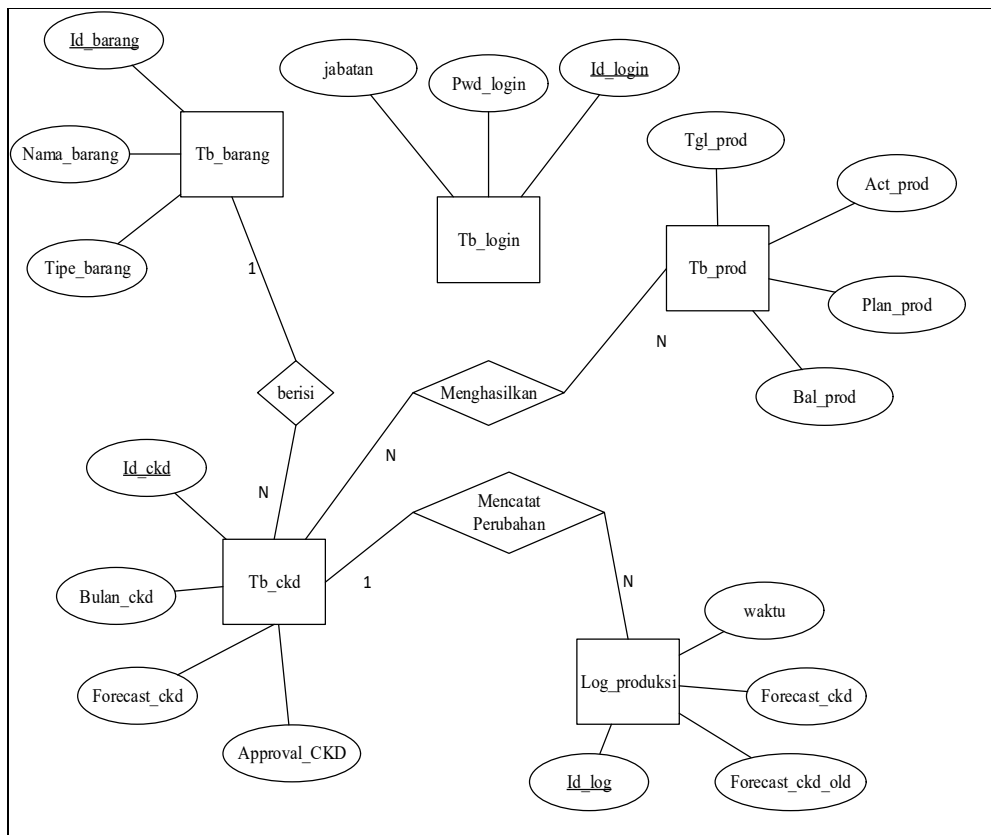
Gambar V.21 *Windows Navigation Diagram*
 Sistem Informasi
 Penjadwalan Produksi
 Sumber: Hasil Analisis(2018)

5.9 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi penjadwalan produksi di PT Krama Yudha Ratu Motor menggunakan dua cara, yakni *Entity Relationship Diagram* (ERD) yang digunakan untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi dari *database* yang digunakan dalam sistem usulan yang akan dibuat.

5.9.1 *Entity Relationship Diagram*(ERD)

Entity relationship diagram dapat membantu dalam merancang hubungan antara entitas dari sebuah *database*. Diagram ini berisi data-data yang berada di dalam *database* yang dapat membantu dalam mengetahui relasi keterhubungan antar tabel. *Entity relationship diagram* sistem informasi penjadwalan produksi dapat dilihat sebagai berikut:



Gambar V.22 *Entity Relationship Diagram* Sistem Informasi Penjadwalan Produksi

Sumber: Hasil Analisis(2018)

5.9.2 Kamus Data

Kamus data membantu dalam pendefinisian data agar pendefinisian data tersebut dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Berikut ini adalah kamus data yang terdapat dalam aplikasi sistem informasi pengiriman *part*:

1. Spesifikasi Tabel Data *Login*

Nama Tabel : *tb_login*

Fungsi : Untuk menyimpan data *login*

Tipe : *File data master*

Tabel V.10 Tabel Data *Login*

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	<i>Username Login</i>	<i>id_login</i>	<i>varchar</i>	8	<i>Primary key</i>
2	<i>Password Login</i>	<i>pwd_login</i>	<i>Varchar</i>	30	
3	Jabatan	<i>jabatan</i>	<i>Varchar</i>	30	

(Sumber: Hasil Analisis Data, 2018)

2. Spesifikasi Tabel Data *Barang*

Nama Tabel : *tb_barang*

Fungsi : Untuk menyimpan data *barang*

Tipe : *File data master*

Tabel V.11 Tabel Data *Barang*

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Kode barang	<i>Id_barang</i>	<i>Integer</i>	8	<i>Primary key</i>
2	Nama barang	<i>Nama Barang</i>	<i>Varchar</i>	30	
3	Tipe barang	<i>Tipe_barang</i>	<i>Varchar</i>	30	

(Sumber: Hasil Analisis Data, 2018)

3. Spesifikasi Tabel Data *CKD Simulation*

Nama Tabel : tb_ckd
 Fungsi : Untuk menyimpan data CKD *Simulation(forecast order)*
 Tipe : *File data transaksi*

Tabel V.12 Tabel Data CKD *Simulation*

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Kode CKD	id_ckd	<i>Integer</i>	8	<i>Primary key</i>
2	Kode barang	id_barang	<i>Integer</i>	8	<i>Foreign key</i>
3	Bulan pemilihan	bulantahun_ckd	<i>Date</i>		
4	Jumlah <i>forecast order</i>	forecast_ckd	<i>Integer</i>	8	
5	Validasi CKD	approval_ckd	<i>varchar</i>	20	

(Sumber: Hasil Analisis Data, 2018)

4. Spesifikasi Tabel Data Produksi

Nama Tabel : tb_produksi
 Fungsi : Untuk menyimpan data produksi
 Tipe : *File data transaksi*

Tabel V.13 Tabel Data Produksi

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Kode Produksi	id_prod	<i>Integer</i>	8	<i>Primary key</i>
2	Kode CKD	id_ckd	<i>Integer</i>	8	<i>Foreign key</i>
3	Hari produksi	tgl_prod	<i>date</i>	-	
4	Jumlah produksi dari CKD	plan_prod	<i>Integer</i>	8	
5	Jumlah produksi yang terjadi	act_prod	<i>Integer</i>	8	
6	<i>Balance</i> produksi	bal_prod	<i>Integer</i>	8	

(Sumber: Hasil Analisis Data, 2018)

5. Spesifikasi Tabel Data Perubahan *Forecast Order*

Nama Tabel : log_produksi
 Fungsi : Untuk menyimpan perubahan data *forecast order*
 Tipe : *File data transaksi*

Tabel V.14 Tabel Data Log Produksi

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1	Id Perubahan	id_log	Integer	8	Primary key
2	Kode CKD	id_ckd	Integer	8	Foreign key
3	Forecast CKD yang Baru	forecast_ckd	Integer	8	
4	Forecast CKD sebelumnya	forecast_ckd_old	Integer	8	
5	Waktu perubahan	waktu	Date	-	

(Sumber: Hasil Analisis Data, 2018)

5.10 Perancangan *Interface* Sistem Informasi Penjadwalan Produksi

Perancangan *interface* merupakan tahapan untuk membuat tampilan atau *design* dari Sistem Informasi Penjadwalan Produksi. Rancangan tampilan yang dibuat meliputi beberapa rancangan *input* (berupa *form input*) dan rancangan *output* (berupa laporan). Rancangan *interface* dalam Sistem Informasi Penjadwalan Produksi tersebut dapat dilihat berikut ini:

1. *Form Login*

Form Login digunakan untuk membedakan hak akses *user*. Melalui *form* ini, pengguna yang dapat masuk kedalam sistem adalah pengguna yang mengetahui kode *login* dan *password* pengguna yang memiliki wewenang untuk menggunakan sistem informasi penjadwalan produksi.

The image shows a login form titled "Krama Yudha Ratu Motor". It features two input fields: "Username" and "Password", each with a placeholder text of the same name. Below the fields is a "Login" button.

Gambar V.23 Rancangan *Form Login*
Sumber: Hasil Analisis(2018)

Keterangan:

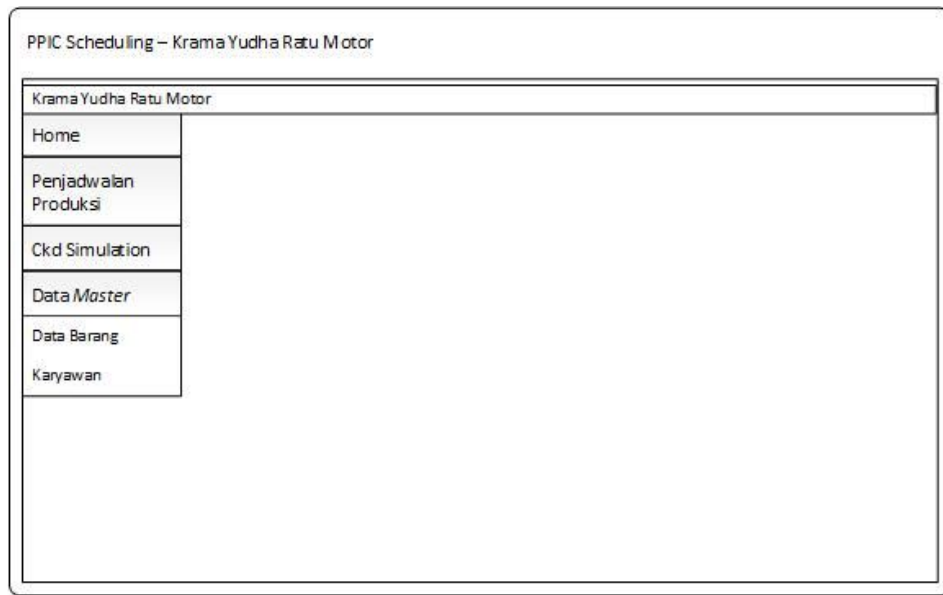
Username : untuk memasukkan kode karyawan

Password : untuk memasukkan *password*

Button Login : Digunakan untuk proses masuk kedalam sistem.

2. Menu Utama

Menu utama ini dirancang untuk memudahkan *user* dalam mengakses aplikasi dengan pilihan-pilihan menu yang sudah disediakan sesuai dengan kebutuhan dan hak akses karena tidak semua menu dapat digunakan.



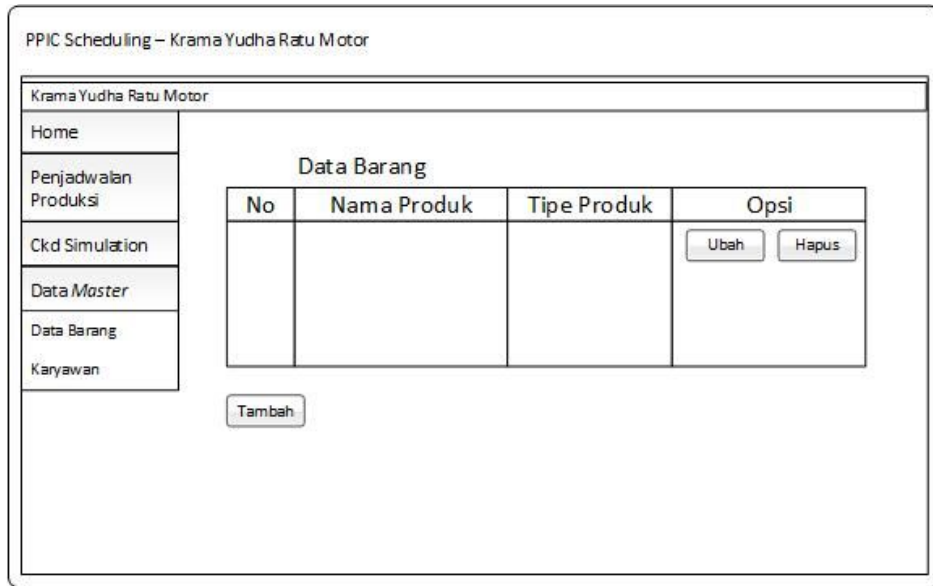
Gambar V.24 Rancangan Menu Utama

Sumber: Hasil Analisis(2018)

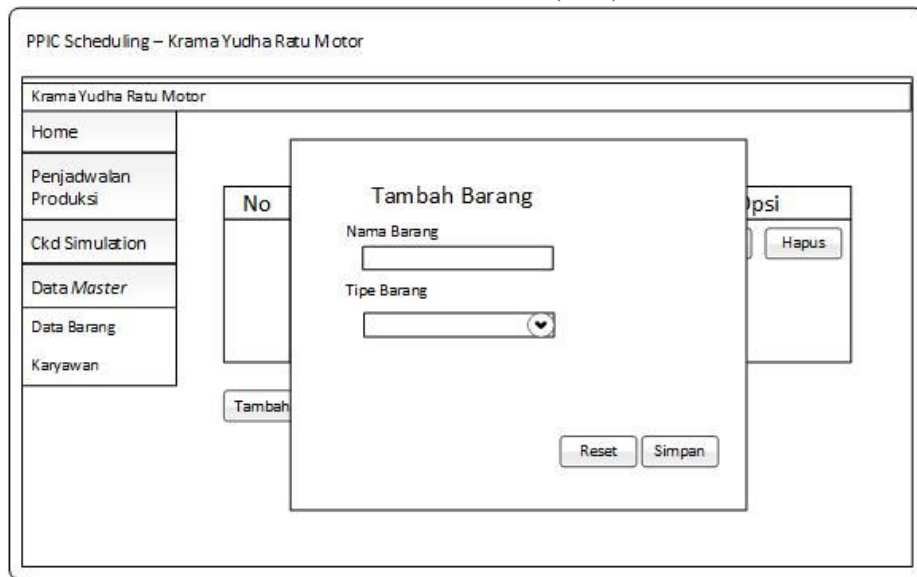
3. Data Master

Data *master* ini terdiri dari *Master* barang, dan data *login*. Berikut penjelasan dari setiap data tersebut:

- 1) *Master* data barang merupakan *form master* yang digunakan untuk melihat dan memasukkan data *master* barang.



Gambar V.25 Rancangan Data Barang
 Sumber: Hasil Analisis(2018)



Gambar V.26 Rancangan *Form* Tambah Data Barang
 Sumber: Hasil Analisis(2018)

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

Home

Penjadwalan Produksi

Ckd Simulation

Data Master

Data Barang

Karyawan

No

Ubah Data Barang

Nama Barang

Tipe Barang

Tambah

Simpan

Hapus

Gambar V.27 Rancangan *Form* Ubah Barang
 Sumber: Hasil Analisis(2018)

2) *Master* data login merupakan *form master* yang digunakan untuk melihat dan memasukkan data *login* yang berfungsi untuk hak akses.

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

Home

Penjadwalan Produksi

Ckd Simulation

Data Master

Data Barang

Karyawan

Data Login

No	Id Karyawan	Password	Jabatan	Ops i
				Ubah Hapus

Tambah

Gambar V.28 Rancangan Data *Login*
 Sumber: Hasil Analisis(2018)

Gambar V.29 Rancangan *Form Tambah Data Login*
 Sumber: Hasil Analisis(2018)

Gambar V.30 Rancangan *Form Ubah Login*
 Sumber: Hasil Analisis(2018)

4. Data CKD *Simulation*

Data CKD *Simulation* adalah *form* data untuk menunjukkan *forecast order* yang ada dan memasukkan data tersebut.

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

- Home
- Penjadwalan Produksi
- Ckd Simulation
- Data Master
- Data Barang
- Karyawan

Data CKD

No	Nama Barang	Bulan	Forecast CKD	Opsi
				Ubah Hapus

Tambah

Gambar V.31 Rancangan Data CKD
 Sumber: Hasil Analisis(2018)

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

- Home
- Penjadwalan Produksi
- Ckd Simulation
- Data Master
- Data Barang
- Karyawan

Tambah Data CKD Simulation

No

Nama Barang

Bulan

Forecast

Opsi Hapus

Tambah

Reset Simpan

Gambar V.32 Rancangan Form Tambah Data CKD
 Sumber: Hasil Analisis(2018)

Gambar V.33 Rancangan *Form* Ubah Data CKD

Sumber: Hasil Analisis(2018)

5. Data Produksi

Form data produksi merupakan data transaksi yang berfungsi untuk memasukkan data produksi yang diramal dan yang terjadi dan membuat laporan

No	Nama Barang	Tanggal	Plan	Actual	Saldo	ACC	Opsi
							Ubah Hapus Validasi

Gambar V.34 Rancangan Data Produksi

Sumber: Hasil Analisis(2018)

Gambar V.35 Rancangan *Form* Tambah Data Produksi
 Sumber: Hasil Analisis(2018)

Gambar V.36 Rancangan *Form* Ubah Data Produksi
 Sumber: Hasil Analisis(2018)

6. Rekap Produksi

Interface rekap produksi merupakan kumpulan dari data produksi yang telah dimasukkan. Data yang dapat dilihat ada data per 1 bulan.

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

Home

Penjadwalan Produksi

Ckd Simulation

Data Master

Data Barang

Karyawan

No

Nama Barang

Bulan

Tambah

Hapus

Rekap

Gambar V.37 Rancangan *Form* Rekap Produksi
 Sumber: Hasil Analisis(2018)

PPIC Scheduling – Krama Yudha Ratu Motor

Krama Yudha Ratu Motor

Home

Penjadwalan Produksi

Ckd Simulation

Data Master

Data Barang

Karyawan

Rekap Produksi

Nama Barang

Bulan

Tanggal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Plan																															
Actual																															
Balance																															

Cetak

Gambar V.38 Rancangan Informasi Rekap Produksi
 Sumber: Hasil Analisis(2018)

7. Cetak Laporan Produksi

Interface ini berfungsi untuk memberikan *output* yang berupa laporan dan dapat melakukan *print-out*, maupun cetak dalam *format *.pdf*

Krama Yudha Ratu Motor

Nama Barang																															
Bulan																															
Tanggal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Plan																															
Actual																															
Balance																															

Cetak

Gambar V.39 Rancangan Laporan Produksi
 Sumber: Hasil Analisis(2018)

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil sistem informasi yang telah dirancang mengenai sistem informasi penjadwalan produksi pada PT Krama Yudha Ratu Motor, dapat disimpulkan beberapa hal, yaitu:

1. Dengan adanya sistem informasi penjadwalan produksi diharapkan dapat membantu dalam proses penyimpanan data penjadwalan produksi, sehingga penyimpanan data dalam 1 lokasi dapat mempermudah pengguna dalam proses pencarian dan penyimpanan data.
2. Sistem yang telah dibangun ini diharapkan dapat membantu dalam mengurangi kesalahan jadwal produksi dengan perhitungan otomatis penjadwalan produksi yang dilakukan oleh sistem.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem informasi penjadwalan produksi ini selanjutnya yaitu:

1. Mengimplementasikan hasil dari penelitian berupa sistem informasi untuk proses penjadwalan produksi *Colt T120ss* di divisi PPIC pada PT Krama Yudha Ratu Motor.
2. Untuk pengembangan sistem informasi, untuk penelitian selanjutnya diharapkan dapat mencakup lebih dari satu produksi.

DAFTAR PUSTAKA

- Anggadini, Sri Dewi. 2011. *Sistem Informasi Akuntansi*. Yogyakarta: Graha Ilmu
- Assauri, Sofjan. 2008. *Manajemen Produksi dan Operasi*. Jakarta: Fakultas Ekonomi UI.
- Dennis, Alan. Barbara Haley, Wixom, Tegarden, David. 2015. *Systems Analysis & Design An Object-Oriented Approach with UML*. Danvers: John Wiley & Sons, Inc.
- Dennis, Alan. 2010. *System Analysis and Design with UML 2.0*. New York: John Wiley and Sons, Inc.
- Et Al, Dennis. 2012 . *Systems Analysis & Design With UML Version 2.0; An Object-Oriented Approach 4th Edition*. John Wiley & Sons
- Gasperz, V. 2008. *Total Quality Management*. Jakarta: Gramedia Pustaka Utama.
- H.M, Jogiyanto. 2010. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Rizky, Soetam. 2011. *Konsep Dasar Rekayasa Perangkat Lunak {Software Rengineering}*. Jakarta: Prestasi Pustaka Publisher.
- Patton, Ron. 2005. *Software Testing(2nd Edition)*: Sams Publishing
- Rusdiana. 2014. *Sistem Informasi Manajemen*. Bandung: Pustaka Setia
- Subagyo, Pangestu. 2006. *Statistik Induktif*. Yogyakarta: BPFE-UGM.
- Sutabri, Tata. 2012. *Analisis Sistem Informasi*. Yogyakarta: Andi.
- Sutaji, Deni. *Sistem Inventory Mini Market dengan PHP & Jquery*. Yogyakarta: Lokomedia.
- Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.

1.17 Jurnal:

- 1.18 Hartono, Iwan. 2014. Aplikasi Penjadwalan Produksi Manufaktur Berbasis Web dengan Metode Master Production Schedule.**
- 1.19 Supriyadi, Riskiyadi. 2016. Penjadwalan Produksi Iks-Filler Pada Proses Ground Calcium Carbonate Menggunakan Metode MPS di Perusahaan Kertas**

LAMPIRAN

barang.php
<?php

```

include "models/mbarang.php";
$brg = new barang($connection);

if(@$_GET['act'] == " "){
?>
<div class="row">
  <div class="col-lg-12">
    <h1>Data Barang</h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="icon-dashboard"></i> Data Barang</a></li>
    </ol>
  </div>
</div>
<div class="row">
  <div class="col-lg-12">
    <div class="table-responsive">
      <table class="table table-hover table-bordered table-striped table-
condensed">
        <tr>
          <th class="col-xs-1" align="center"><center>No.</center></th>
          <!-- <th>Kode Produk</th> -->
          <th class="col-xs-6">Nama Produk</th>
          <th class="col-xs-3">Tipe Produk</th>
          <th class="col-xs-2" align="center"><center>Ops</center></th>
        </tr>
      <?php
        $sno = 1;
        $stampil = $brg->tampil();
        while ($data = $stampil->fetch_object()) { ?>
          <tr>
            <td align="center"><?php echo $sno++; ?>. </td>
            <!-- <td><?php echo $data->id_barang; ?></td> -->
            <td><?php echo $data->nama_barang; ?></td>
            <td><?php echo $data->tipe_barang; ?></td>
            <td align="center">
              <a id="edit_barang" data-toggle="modal" data-target="#edit"
                data-id= "<?php echo $data->id_barang; ?>"
                data-nama= "<?php echo $data->nama_barang; ?>"
                data-tipe= "<?php echo $data->tipe_barang; ?>">
                <button
class="btn btn-primary btn-xs"><i class="fa fa-edit"></i> Ubah</button>
              </a>
              <a href="?page=barang&act=del&id=<?php echo $data-
>id_barang;?>" onclick="return confirm('Apakah anda yakin ingin menghapus data
ini?')">
                <button type="button" class="btn btn-danger btn-xs"><i class="fa fa-
trash-o"></i> Hapus</button></a>
            </td>
          </tr>
        <?php
        } ?>

```

```

</table> </div>
<button type="button" class="btn btn-success" data-toggle="modal" data-
target="#tambah">Tambah</button>
<div id="tambah" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Tambah Data Barang </h4>
      </div>
      <form action="" method="post" enctype="multipart/form-data">
        <div class="modal-body">
          <div class="form-group">
            <label class="control-label" for="nama_barang">Nama Barang</label>
            <input type="text" name="nama_barang" class="form-control"
id="nama_barang" required>
            <label class="control-label" for="tipe_barang">Tipe Barang</label>
            <select class="form-control" name="tipe_barang" id="tipe_barang"
required>
              <option>Standard</option>
              <option>Flat Bed</option>
              <option>Three Way</option>
            </select>
          </div>
          </div>
          <div class="modal-footer">
            <button type="reset" class="btn btn-danger">Reset</button>
            <button type="submit" class="btn btn-success" name="tambah"
value="Simpan">Simpan</button>
          </div>
        </form>
        <?php
          if(@$_POST['tambah']){
            $nama_barang = $connection->conn-
>real_escape_string($_POST['nama_barang']);
            $tipe_barang = $connection->conn-
>real_escape_string($_POST['tipe_barang']);
            $brg->tambah($nama_barang, $tipe_barang);
            header("location: ?page=barang");
          }
        ?>
      </div>
    </div>
  </div>
  <div id="edit" class="modal fade" role="dialog">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">

```

```

        <button type="button" class="close" data-
dismiss="modal">&times;</button>
        <h4 class="modal-title">Ubah Data Barang </h4>
        </div>
        <form id="form" action="" method="post"
enctype="multipart/form-data">
        <div class="modal-body" id="modal-edit">
        <div class="form-group">
        <label class="control-label" for="id_barang">Kode
Barang</label>
        <input type="text" name="id_barang" class="form-control"
id="id_barang" readonly>
        <label class="control-label" for="nama_barang">Nama
Barang</label>
        <input type="text" name="nama_barang" class="form-control"
id="nama_barang" required>
        <label class="control-label" for="tipe_barang">Tipe
Barang</label>
        <select class="form-control" name="tipe_barang"
id="tipe_barang" required>
        <option>Standard</option>
        <option>Flat Bed</option>
        <option>Three Way</option>
        </select>
        </div>
        </div>
        <div class="modal-footer">
        <button type="submit" class="btn btn-success" name="edit"
value="ubah">Ubah</button>
        </div>
        </form>
        <?php
        if(@$_POST['edit']){
            $id_barang = $connection->conn-
>real_escape_string($_POST['id_barang']);
            $nama_barang = $connection->conn-
>real_escape_string($_POST['nama_barang']);
            $tipe_barang = $connection->conn-
>real_escape_string($_POST['tipe_barang']);
            $brg->edit($id_barang, $nama_barang, $tipe_barang);
            header("location: ?page=barang");
        }
        ?>
        </div>
    </div>
    </div>

<script src="assets/js/jquery-1.10.2.js"></script>
<script type="text/javascript">
    $(document).on("click", "#edit_barang", function() {

```

```

        var idbarang = $(this).data('id');
        var namabarang = $(this).data('nama');
        var tipebarang = $(this).data('tipe');
        $("#modal-edit #id_barang").val(idbarang);
        $("#modal-edit #nama_barang").val(namabarang);
        $("#modal-edit #tipe_barang").val(tipebarang);
    })
</script>
</div>
</div>
<?php
}
else if(@$_GET['act'] == 'del'){
    $brg->delete($_GET['id']);
    header("location: ?page=barang");
}
}

```

mbarang.php

```

<?php
class barang{
    private $mysqli;

    function __construct($conn){
        $this->mysqli = $conn;
    }

    public function tampil($id = null){
        $db = $this->mysqli->conn;
        $sql = "SELECT * FROM tb_barang";
        if($id != null){
            $sql .= " WHERE id_barang = $id";
        } else{
            $sql .= "";
        }
        $query = $db->query($sql) or die ($db->error);
        return $query;
    }

    public function tambah($nama_barang, $tipe_barang){
        $db = $this->mysqli->conn;
        $db->query("INSERT INTO tb_barang VALUES(',$nama_barang',
'$tipe_barang')") or die($db->error);
    }

    public function edit($id_barang, $nama_barang, $tipe_barang){
        $db = $this->mysqli->conn;
        $db->query("UPDATE tb_barang SET nama_barang = '$nama_barang',
tipe_barang = '$tipe_barang' WHERE id_barang = '$id_barang' ") or die($db->error);
    }
}

```

```

    public function delete($id){
        $db = $this->mysqli->conn;
        $db->query("DELETE FROM tb_barang WHERE id_barang = '$id'") or
die($db->error);
    }
}
?>

```

datalogin.php

```

<?php
include "models/mlogin.php";

$loginclass = new login($connection);

if(@$_GET['act'] == " "){
?>

<div class="row">
    <div class="col-lg-12">
        <h1>Data Login Karyawan</h1>
        <ol class="breadcrumb">
            <li><a href="#"><i class="icon-dashboard"></i> Data Login
Karyawan</a></li>
        </ol>
    </div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="table-responsive">
            <table class="table table-hover table-bordered table-striped table-
condensed">
                <tr>
                    <th class="col-xs-1" align="center"><center>No.</center></th>
                    <!-- <th>Kode Produk</th> -->
                    <th>ID Login</th>
                    <th>Password</th>
                    <th>Nama Karyawan</th>
                    <th>Hak Akses</th>
                    <th class="col-xs-2"
align="center"><center>Ops</center></th>
                </tr>
                <?php
                $no = 1;
                $stampil = $loginclass->tampil();
                while ($data = $stampil->fetch_object()) { ?>
                <tr>
                    <td align="center"><?php echo $no++; ?>. </td>
                    <td><?php echo $data->id_login; ?></td>
                    <td><?php echo $data->pwd_login; ?></td>

```

```

<td><?php echo $data->nama_karyawan; ?></td>
<td><?php echo $data->hak_akses; ?></td>
<td align="center">
    <a id="edit_login" data-toggle="modal" data-
target="#edit"
    data-id= "<?php echo $data->id_login; ?>"
    data-pwd= "<?php echo $data->pwd_login; ?>"
    data-nama= "<?php echo $data->nama_karyawan; ?>"
    data-hakakses= "<?php echo $data->hak_akses; ?>"
    <button class="btn btn-primary btn-xs"><i
class="fa fa-edit"></i> Ubah</button>
    </a>
    <a href="?page=datalogin&act=del&id=<?php echo
$data->id_login;?>" onclick="return confirm('Apakah anda yakin ingin menghapus data
ini?')">
    <button type="button" class="btn btn-danger btn-xs"><i
class="fa fa-trash-o"></i> Hapus</button></a>
</td>
</tr>
<?php
} ?>
</table>
</div>
<button type="button" class="btn btn-success" data-toggle="modal" data-
target="#tambah">Tambah</button>
<div id="tambah" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal">&times;</button>
                <h4 class="modal-title">Tambah Data
Login</h4>
            </div>
            <form action="" method="post"
enctype="multipart/form-data">
                <div class="modal-body">
                    <div class="form-group">
                        <label class="control-label"
for="id_login">ID Karyawan</label>
                        <input type="text"
name="id_login" class="form-control" id="id_login" required>
                        <label class="control-label"
for="pwd_login">Password Karyawan</label>
                        <input type="text"
name="pwd_login" class="form-control" id="pwd_login" required>
                        <label class="control-label"
for="nama_karyawan">Nama Karyawan</label>
                        <input type="text"
name="nama_karyawan" class="form-control" id="nama_karyawan" required>

```

```

<label class="control-label"
for="hak_akses">Hak Akses</label>
<select class="form-control"
name="hak_akses" id="hak_akses" required>
    <option>admin</option>
    <option>kabag</option>
    <option>pegawai</option>
    <option>foreman</option>
</select>
</div>
<div class="modal-footer">
    <button type="reset" class="btn btn-
danger">Reset</button>
    <button type="submit" class="btn btn-
success" name="tambah" value="Simpan">Simpan</button>
</div>
</form>
<?php
if(@$_POST['tambah']){
    $id_login = $connection->conn-
>real_escape_string($_POST['id_login']);
    $pwd_login = $connection->conn-
>real_escape_string($_POST['pwd_login']);
    $nama_karyawan = $connection->conn-
>real_escape_string($_POST['nama_karyawan']);
    $hak_akses= $connection->conn-
>real_escape_string($_POST['hak_akses']);
    $loginclass->tambah($id_login, $pwd_login,
$nama_karyawan, $hak_akses);
    header("location: ?page=datalogin");
}
?>
</div>
</div>
<div id="edit" class="modal fade" role="dialog">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal">&times;</button>
                <h4 class="modal-title">Ubah Data Login </h4>
            </div>
            <form id="form" action="" method="post"
enctype="multipart/form-data">
                <div class="modal-body" id="modal-edit">
                    <div class="form-group">

```

```

        <label class="control-label"
for="id_login">ID Karyawan</label>
        <input type="text"
name="id_login" class="form-control" id="id_login" required readonly>
        <label class="control-label"
for="pwd_login">Password Karyawan</label>
        <input type="text"
name="pwd_login" class="form-control" id="pwd_login" required>
        <label class="control-label"
for="nama_karyawan">Nama Karyawan</label>
        <input type="text"
name="nama_karyawan" class="form-control" id="nama_karyawan" required>
        <label class="control-label"
for="hak_akses">Hak Akses</label>
        <select class="form-control"
name="hak_akses" id="hak_akses" required>
                <option>admin</option>
                <option>kabag</option>
                <option>pegawai</option>
                <option>foreman</option>
        </select>
    </div>
</div>
<div class="modal-footer">
    <button type="submit" class="btn btn-
success" name="edit" value="ubah">Ubah</button>
</div>
</form>
<?php
if(@$_POST['edit']){
    $id_login = $connection->conn-
>real_escape_string($_POST['id_login']);
    $pwd_login = $connection->conn-
>real_escape_string($_POST['pwd_login']);
    $nama_karyawan = $connection->conn-
>real_escape_string($_POST['nama_karyawan']);
    $hak_akses= $connection->conn-
>real_escape_string($_POST['hak_akses']);
    $loginclass->edit($id_login, $pwd_login,
$nama_karyawan, $hak_akses);
    header("location: ?page=datalogin");
}
?>
</div>
</div>
</div>
<script src="assets/js/jquery-1.10.2.js"></script>
<script type="text/javascript">
    $(document).on("click", "#edit_login", function() {

```

```

        var idlogin = $(this).data('id');
        var pwdlogin = $(this).data('pwd');
        var namallogin = $(this).data('nama');
        var hakakses = $(this).data('hakakses');
        $("#modal-edit #id_login").val(idlogin);
        $("#modal-edit #pwd_login").val(pwdlogin);
        $("#modal-edit #nama_karyawan").val(namallogin);
        $("#modal-edit #hak_akses").val(hakakses);
    })
</script>
</div>
</div>
<?php
}
else if(@$_GET['act'] == 'del'){
    $loginclass->delete($_GET['id']);
    header("location: ?page=datalogin");
}

```

mlogin.php

```

<?php
class login{
    private $mysqli;

    function __construct($conn){
        $this->mysqli = $conn;
    }

    public function tampil($id = null){
        $db = $this->mysqli->conn;
        $sql = "SELECT * FROM tb_login";
        if($id != null){
            $sql .= " WHERE id_login = $id";
        } else{
            $sql .= "";
        }
        $query = $db->query($sql) or die ($db->error);
        return $query;
    }

    public function tambah($id_login, $pwd_login, $nama_karyawan, $hak_akses){
        $db = $this->mysqli->conn;
        $db->query("INSERT INTO tb_login VALUES('$id_login', '$pwd_login',
'$nama_karyawan', '$hak_akses')") or die($db->error);
    }

    public function edit($id_login, $pwd_login, $nama_karyawan, $hak_akses){
        $db = $this->mysqli->conn;

```

```
        $db->query("UPDATE tb_login SET pwd_login = '$pwd_login', hak_akses =
'$hak_akses', nama_karyawan = '$nama_karyawan' WHERE id_login = '$id_login' ") or
die($db->error);
    }
    public function delete($id){
        $db = $this->mysql->conn;
        $db->query("DELETE FROM tb_login WHERE id_login = '$id'") or die($db-
>error);
    }
}
```